

UNIVERSIDAD CARLOS III DE MADRID

ESCUELA POLITÉCNICA SUPERIOR

INGENIERÍA TÉCNICA INFORMÁTICA DE GESTIÓN



PROYECTO FIN DE CARRERA

PGS: PROCESO DE GESTIÓN DE SALENDA

AUTOR: ROBERTO MARTÍN FLORES

TUTOR: ALEJANDRO CALDERÓN MATEOS

26 de Septiembre de 2015

Título: Proceso de Gestión de Salenda

Autor: Roberto Martín Flores

Director: Alejandro Calderón Mateos

EL TRIBUNAL

Presidente: *****

Vocal: *****

Secretario *****

Realizado el acto de defensa y lectura del Proyecto Fin de Carrera el día XX de XXXXXX de 2015 en Leganés, en la Escuela Politécnica Superior de la Universidad Carlos III de Madrid, acuerda otorgarle la CALIFICACIÓN de

VOCAL

SECRETARIO

PRESIDENTE

Agradecimientos

A mis padres, por darme la posibilidad de tener unos estudios que me han permitido acceder a una profesión que respeto, amo y disfruto cada día.

A mi hermano, por ser un apoyo siempre, y ayudarme con su experiencia y cariño, a superar las primeras etapas de esta carrera, que a veces, se hace cuesta arriba.

A mi amigo y jefe, Álvaro Sanchez-Mariscal, por darme la oportunidad, no solo de realizar este proyecto, si no de embarcarme en un viaje, que profesional y personalmente, me ha dado ya 5 grandes años de experiencia y realización profesional.

A mi tutor, Alejandro Calderón Mateos, por sus consejos y paciencia.

Y por último a Vanesa Tejada Muñoz, el amor de mi vida, mi mujer y la persona responsable de que este proyecto haya llegado a su fin: gracias a su amor, perseverancia, paciencia perdida y recuperada una y mil veces, sufrimiento y sobre todo motivación y apoyo, ha conseguido, que 4 años después de finalizado este proyecto, completara su documentación, y consiguiera ser finalmente, un ingeniero titulado, jamás te estaré suficientemente agradecido, te quiero, eres mi vida.

*El futuro tiene muchos nombres. Para los débiles es lo inalcanzable. Para los temerosos,
lo desconocido. Para los valientes es la oportunidad.*

Victor Hugo

Un gran poder conlleva siempre una gran responsabilidad

Franklin D. Roosevelt

Resumen

Proceso de gestión de Salenda [\[1\]](#) , es una aplicación que facilita el trabajo de un empresario para conocer el estado de sus proyectos, empleados, cobros y presupuestos de clientes, con la finalidad de poder manejar los recursos y llevar a cabo una mejor toma de decisiones de su negocio.

La experiencia de poder desarrollar una herramienta, de gestión de procesos interna propia de una empresa, permite comprender la complejidad de la gestión de un negocio, y la dificultad de la implementación de un sistema de backend, donde se puedan controlar los recursos activos y pasivos con todo detalle.

Palabras clave: Aplicación web, empresa, proyectos, gestión, grails, java.

Abstract

Salenda's [\[1\]](#) Processes Management, is web application that facilitates the work of an entrepreneur to know the status of their projects costs and revenue, staff salaries and costs, invoices and budgets of clients, in order to be able to manage resources and take improved bussiness decision.

The opportunity to develop an internal bussiness management application for a small startup, allows us to understand the complexity of managing a consulting business, and the difficulty of the implementation of a backend system where you can control the assets and liabilities in an all-in-one application.

Keywords: Web application, enterprise, bussiness, project management, rails, java.

Índice general

1. Introducción y Objetivos	25
1.1. Introducción	27
1.2. Objetivos	27
1.2.1. Herramienta de gestión de proyectos de software	27
1.2.2. Herramientas ágiles de desarrollo	28
1.3. Fases del Desarrollo	29
1.4. Medios empleados	29
1.5. Estructura de la memoria	30
2. Estado del Arte	33
2.1. Salenda	35
2.2. Soluciones existentes	35
2.2.1. Jira	36
2.2.2. SugarCRM	37
2.2.3. SAGE. ContaPlus, FacturaPlus, NominaPlus	39
2.2.4. Comparativa	40
2.3. Tecnologías de desarrollo de Software	42
2.3.1. Herramientas estandarizadas. Java + J2EE	42
2.3.2. Nuevas tecnologías. Groovy + Grails	43
2.3.3. Java vs Groovy	45
2.4. Metodologías de desarrollo de Software	47

2.4.1. Evolución histórica	47
2.4.2. Scrum	50
2.5. Prueba de concepto	55
3. Análisis	59
3.1. Identificación de Requisitos	61
3.1.1. Plantilla	61
3.1.2. Requisitos Funcionales	62
3.1.3. Requisitos No Funcionales	82
3.2. Identificación de Usuarios	85
3.2.1. Recursos Humanos	85
3.2.2. Contable/Administrativo	85
3.2.3. Técnico Preventa	86
3.2.4. Ingeniero de Software	86
3.2.5. Administrador	86
3.3. Diagramas de Casos de Uso	86
3.3.1. Plantilla	87
3.3.2. Caso de uso general	89
3.3.3. Casos de uso del módulo de empleados	90
3.3.4. Casos de uso del módulo de compras	93
3.3.5. Casos de uso del módulo de ventas	96
3.3.6. Casos de uso del módulo de proyectos	98
3.3.7. Casos de uso del módulo de balance	100
3.3.8. Casos de uso del sistema	102
4. Diseño	105
4.1. Arquitectura de Software	107
4.1.1. Modelo Cliente - Servidor	107
4.1.2. Modelo-Vista-Controlador	108
4.2. Diseño del Modelo de datos	111
4.2.1. Diagrama de Entidad-Relación	111

4.2.2.	Modelo relacional	116
4.3.	Diseño de la Interfaz de usuario	117
4.3.1.	Módulo de empleados	117
4.3.2.	Módulo de compras	120
4.3.3.	Módulo de ventas	122
4.3.4.	Módulo de proyectos	124
4.3.5.	Módulo de balance	126
4.3.6.	Módulo de configuración	129
5.	Planificación y Presupuesto	131
5.1.	Planificación del proyecto	133
5.2.	Diagrama de Gantt	134
5.3.	Presupuesto	136
5.3.1.	Costes de personal	136
5.3.2.	Costes de materiales	137
5.3.3.	Costes de funcionamiento	137
5.3.4.	Resumen	138
6.	Implantación y manual de usuario	141
6.1.	Implantación	143
6.1.1.	Instalación de Java6: java	143
6.1.2.	Instalación de Java6: java home	144
6.1.3.	Instalación de Tomcat 5.5	145
6.1.4.	Instalación de MySQL Server 5.1	146
6.1.5.	Despliegue	146
6.2.	Manual de usuario	147
6.2.1.	Panel de Empleados	147
6.2.2.	Panel de Compras	153
6.2.3.	Panel de Ventas	158
6.2.4.	Panel de Proyectos	162
6.2.5.	Panel de Balance	167

6.2.6. Panel de Configuración	175
---	-----

7. Conclusiones y Líneas Futuras	179
---	------------

7.1. Conclusiones	181
7.1.1. Herramienta de gestión de proyectos de software	181
7.1.2. Herramientas ágiles de desarrollo	182
7.2. Líneas Futuras	184

Lista de Figuras

2.1. FileReader.java	45
2.2. FileReader.groovy	45
2.3. Saludador.java	46
2.4. Saludador.groovy	47
2.5. Ciclo de vida del software	48
2.6. Scrum Roles	52
2.7. Scrum Meetings	53
3.1. Diagramas de casos de uso: Representación de actor	87
3.2. Diagramas de casos de uso: Representación de caso de uso	87
3.3. Diagramas de casos de uso: Representación de relación de asociación	87
3.4. Diagramas de casos de uso: Representación de relación de inclusión	88
3.5. Diagramas de casos de uso: Representación de relación de especialización	88
3.6. Diagramas de casos de uso: Representación de escenario	88
3.7. Diagramas de casos de uso: Caso de uso general	89
3.8. Diagramas de casos de uso: Módulo de empleados	90
3.9. Diagramas de casos de uso: Módulo de compras	93
3.10. Diagramas de casos de uso: Módulo de ventas	96
3.11. Diagramas de casos de uso: Módulo de proyectos	98
3.12. Diagramas de casos de uso: Módulo de balance	100
3.13. Diagramas de casos de uso: Sistema	102

4.1. Modelo cliente-servidor	107
4.2. Patrón Modelo-Vista-Controlador	108
4.3. Grails MVC: /grails-app/controllers/PersonaController.groovy	109
4.4. Grails MVC: /grails-app/domain/Persona.groovy	110
4.5. Grails MVC: /grails-app/views/persona/show.groovy	110
4.6. Diagrama E/R: Entidad	111
4.7. Diagrama E/R: Relación	111
4.8. Diagrama E/R: Restricción	112
4.9. Diagrama E/R: General	112
4.10. Diagrama E/R: Empleados	113
4.11. Diagrama E/R: Compras	113
4.12. Diagrama E/R: Ventas	114
4.13. Diagrama E/R: Proyectos	114
4.14. Diagrama E/R: Balance	115
4.15. Modelo Relacional	116
4.16. Diseños: Pestaña empleados	117
4.17. Diseños: Detalle baja empleado	118
4.18. Diseños: Detalle empleado	118
4.19. Diseños: Detalle paga mensual empleado	119
4.20. Diseños: Pestaña compras	120
4.21. Diseños: Detalle de compra	121
4.22. Diseños: Detalle de proveedor	121
4.23. Diseños: Pestaña ventas	122
4.24. Diseños: Detalle de factura	123
4.25. Diseños: Detalle de cliente	123
4.26. Diseños: Módulo de proyectos	124
4.27. Diseños: Creación de proyecto	124
4.28. Diseños: Detalle de proyecto	125
4.29. Diseños: Detalle de worklog	125
4.30. Diseños: Módulo de balance	126
4.31. Diseños: Detalle de pago	127

4.32. Diseños: Detalle de cobro	127
4.33. Diseños: Detalle de impuesto	128
4.34. Diseños: Módulo de configuración	129
5.1. Diagrama de Gantt	135
5.2. Fórmula del coste de personal	136
5.3. Fórmula del cálculo de la amortización	137
6.1. Instalación de Java	144
6.2. Instalación de Tomcat	145
6.3. Panel de empleados	147
6.4. Creación de empleado	148
6.5. Edición de empleado	149
6.6. Baja de empleado	149
6.7. Pagas mensuales empleado	150
6.8. Creación de paga	151
6.9. Edición de paga	151
6.10. Empleados: Herramientas administrativas	152
6.11. Panel de compras	153
6.12. Graficas de compras	154
6.13. Creación de compra	155
6.14. Edición de compra	155
6.15. Edición de proveedor	156
6.16. Compras: Herramientas administrativas	157
6.17. Panel de ventas	158
6.18. Creación de factura	160
6.19. Edición de factura	161
6.20. Edición de cliente	161
6.21. Panel de proyectos	162
6.22. Grafica de proyectos. Sumario de proyectos	163
6.23. Grafica de proyectos. Situación de proyectos	164
6.24. Creación de proyecto	164

6.25. Edición de proyecto	165
6.26. Creación de worklog	166
6.27. Edición de worklog	167
6.28. Panel de Balance	168
6.29. Panel de Balance: detalle	169
6.30. Creación de pago	170
6.31. Modificación de pago	170
6.32. Creación de cobro	171
6.33. Modificación de cobro	172
6.34. Creación de impuesto	172
6.35. Modificación de impuesto	173
6.36. Balance: Herramientas administrativas	174
6.37. Panel de configuración	175

Lista de Tablas

2.1. Soluciones Existentes: Comparativa	41
3.1. Plantilla de requisitos	61
3.2. RF-1. Módulo de empleados	62
3.3. RF-2. Listado de empleados	62
3.4. RF-3. Alta de empleado	62
3.5. RF-4. Modificación de empleado	63
3.6. RF-5. Baja de empleado	63
3.7. RF-6. Eliminación de empleado	63
3.8. RF-7. Gráfica de coste/hora	63
3.9. RF-8. Cálculo del coste/hora	64
3.10. RF-9. Gráfica de horas/mes	64
3.11. RF-10. Cálculo de las horas/mes	64
3.12. RF-11. Listado de últimas pagas empleado	64
3.13. RF-12. Alta de paga empleado	65
3.14. RF-13. Modificación de paga empleado	65
3.15. RF-14. Eliminación de paga empleado	65
3.16. RF-15. Módulo de compras	66
3.17. RF-16. Listado de proveedores	66
3.18. RF-17. Alta de proveedor	66
3.19. RF-18. Modificación de proveedor	66
3.20. RF-19. Eliminación de proveedor	67

3.21. RF-20. Listado de compras	67
3.22. RF-21. Alta de compra	67
3.23. RF-22. Modificación de compra	68
3.24. RF-23. Eliminación de compra	68
3.25. RF-24. Gráfica de compras	68
3.26. RF-25. Módulo de ventas	69
3.27. RF-26. Listado de clientes	69
3.28. RF-27. Alta de cliente	69
3.29. RF-28. Modificación de cliente	69
3.30. RF-29. Eliminación de cliente	70
3.31. RF-30. Listado de facturas emitidas	70
3.32. RF-31. Listado de facturas pendientes de cobro	70
3.33. RF-32. Alta de factura	71
3.34. RF-33. Modificación de factura	71
3.35. RF-34. Eliminación de factura	72
3.36. RF-35. Gráfica de facturación trimestral	72
3.37. RF-36. Módulo de proyectos	72
3.38. RF-37. Listado de proyectos	72
3.39. RF-38. Alta de proyecto	73
3.40. RF-39. Modificación de proyecto	73
3.41. RF-40. Eliminación de proyecto	73
3.42. RF-41. Cálculo de costes de un proyecto	74
3.43. RF-42. Gráfica de sumario de proyectos	74
3.44. RF-43. Gráfica de situación de proyectos	75
3.45. RF-44. Listado de últimos worklogs	75
3.46. RF-45. Alta de worklog	75
3.47. RF-46. Modificación de worklog	76
3.48. RF-47. Eliminación de worklog	76
3.49. RF-48. Módulo de balance	76
3.50. RF-49. Previsión de pagos	76
3.51. RF-50. Alta de pago	77

3.52. RF-51. Modificación de pago	77
3.53. RF-52. Eliminación de pago	77
3.54. RF-53. Previsión de cobros	78
3.55. RF-54. Alta de cobro	78
3.56. RF-55. Modificación de cobro	78
3.57. RF-56. Eliminación de cobro	79
3.58. RF-57. Previsión de impuestos	79
3.59. RF-58. Alta de impuesto	79
3.60. RF-59. Modificación de impuesto	80
3.61. RF-60. Eliminación de impuesto	80
3.62. RF-61. Gráfica de balance de caja	80
3.63. RF-62. Proceso de cambio de mes	81
3.64. RF-63. Amortizar las compras	81
3.65. RF-64. Copiar pagas mes anterior	82
3.66. RF-65. Generar pagos de nóminas y SS	82
3.67. RF-66. Copiar compras recurrentes	82
3.68. RNF-1. Software multiplataforma	83
3.69. RNF-2. Compatibilidad con navegadores	83
3.70. RNF-3. Disponibilidad on-line	83
3.71. RNF-4. Usabilidad	83
3.72. RNF-5. Multi-idioma	84
3.73. RNF-6. Tolerancia a fallos	84
3.74. RNF-7. Experiencia de usuario	84
3.75. RNF-8. Manual de usuario	84
3.76. RNF-9. Interfaz de usuario	85
3.77. Plantilla de casos de uso	88
3.78. UC-1. Caso de uso general	90
3.79. UC-2. Filtrar empleados	91
3.80. UC-3. Baja empleado	91
3.81. UC-4. Eliminar empleado	92
3.82. UC-5. Gráfica horas/mes	92

3.83. UC-6. Agrupar gráfica	94
3.84. UC-7. Modificación compra	94
3.85. UC-8. Alta pago	95
3.86. UC-9. Agrupar facturación	97
3.87. UC-10. Listar facturas pendientes	97
3.88. UC-11. Actualizar valores	99
3.89. UC-12. Alta worklog	99
3.90. UC-13. Modificar pago	101
3.91. UC-14. Alta impuesto	101
3.92. UC-15. Cambio de mes	102
5.1. Planificación del proyecto: Product backlog	134
5.2. Diagrama de Gantt: PGS	134
5.3. Presupuesto: Gastos de personal	136
5.4. Presupuesto: Gastos de materiales hardware	137
5.5. Presupuesto: Resumen	138

Capítulo 1

Introducción y Objetivos

En este capítulo del documento, se explica brevemente, cuál ha sido la motivación de la realización de este proyecto, cual es la problemática que se pretende resolver, los objetivos principales que se persiguen, así como las fases de desarrollo en las que se ha dividido y un listado de los medios con los que se ha contado para su realización.

Adicionalmente se recoge un breve resumen de la estructura del resto del documento, con el fin de facilitar al lector su estudio y comprensión.

1.1. Introducción

Salenda [1] es una *pyme* [2] española, especializada en el desarrollo de aplicaciones, los servicios de consultoría y la formación técnica, que en su búsqueda de una herramienta de gestión de proyectos, llega a la conclusión de que, actualmente, no existe una única solución que cubra completamente sus necesidades, y las herramientas existentes que ya están utilizando, tienen unos costes de licencia elevados para una pyme.

Partiendo de este escenario y tratándose de una empresa de desarrollo de software, deciden implementar su propia solución, con el fin de dar cobertura a todas sus necesidades en una sola aplicación.

1.2. Objetivos

En esta sección se explican los objetivos que se pretenden alcanzar: El desarrollo de una aplicación de gestión de proyectos de software utilizando metodologías y frameworks de desarrollo ágiles.

1.2.1. Herramienta de gestión de proyectos de software

El objetivo principal, que se pretende alcanzar, es una aplicación de *gestión de proyectos* [3] de desarrollo de software que permita obtener, en tiempo real, el coste y los beneficios de los proyectos que se están acometiendo en cada momento en la empresa, con la finalidad de poder mejorar la toma de decisiones de negocio, y que esto permita alcanzar una ventaja competitiva en un sector muy explotado. Las principales funcionalidades a cubrir por esta aplicación serán:

- Gestión de Salarios y Costes de los *Empleados*
- Gestión de *Compras*, Amortizaciones y Proveedores de servicios y materiales
- Gestión de Clientes, *Ventas* y Facturación
- Gestión de *Proyectos*, Costes y Beneficios
- *Balance* de Caja y Pagos periódicos

1.2.2. Herramientas ágiles de desarrollo

Dado que se trata de un proyecto de financiación propia, se pretende optimizar al máximo los recursos destinados al proyecto. Para ello, se utilizarán metodologías y frameworks [4] que minimicen el tiempo de desarrollo y sus costes.

Frameworks ágiles de desarrollo

Un *framework* [4] de desarrollo, es un conjunto de herramientas (o librerías), estructuras y reglas, que facilitan ciertas etapas del proceso de implementación del software. Aportan soluciones a problemas conocidos y recurrentes como la persistencia de datos, o su presentación, entre otros. Entre ellos, los frameworks ágiles de desarrollo, surgen para permitir una mayor capacidad de adaptación al cambio y evolución de las funcionalidades de una aplicación.

El framework de desarrollo elegido en este caso ha sido *Grails* [5], un framework para aplicaciones web sobre la *JVM* [6] que aprovecha el lenguaje de programación dinámico *Groovy* [7] y se basa en el patrón *Convención sobre Configuración* [8] para proporcionar una experiencia productiva y ágil al programador.

Metodologías ágiles de desarrollo

Una *metodología de desarrollo de software* [9] se refiere a un framework que, en lugar de centrarse en la implementación, se centra en la estructuración, planificación y control del proceso de desarrollo. Existen en la actualidad un gran número de Metodologías de desarrollo diferentes, desde los años 70 hasta nuestro días, las metodologías han evolucionado siguiendo cada una su enfoque particular.

Durante la década de los 90, surgieron por toda la comunidad de desarrollo movimientos que promovían nuevas formas de estructurar el desarrollo, metodologías más ágiles que las anteriores, que permitieran alta flexibilidad y rápida adaptación a los cambios de requisitos.

La metodología de desarrollo elegida para este proyecto ha sido *Scrum* [10], una metodología ágil que evoluciona del modelo iterativo-incremental y se caracteriza por los equipos auto-gestionados y la entrega de valor real en cada iteración.

1.3. Fases del Desarrollo

El desarrollo de esta aplicación se dividió en las siguientes fases:

- Estudio de la situación actual de la empresa y sus necesidades.
- Análisis del estado de las tecnologías de desarrollo de aplicaciones web.
- Definición de objetivos primarios y secundarios.
- Toma de requisitos y especificación, a alto nivel, de la interfaz deseada.
- Planificación y presupuesto.
- Diseño inicial de la aplicación
- Implementación:
 - o Módulo 1: Empleados
 - o Módulo 2: Compras
 - o Módulo 3: Ventas
 - o Módulo 4: Proyectos
 - o Módulo 5: Balance

En cada fase:

Análisis, estimación, desarrollo, pruebas, demo y toma de requisitos para la siguiente iteración.

- Puesta en producción

1.4. Medios empleados

Para el desarrollo de esta aplicación se utilizaron los siguientes medios:

- Ordenador personal con sistema operativo linux
- Entorno de desarrollo [11] *Grail's Tool Suite* [12]

- Servidor de control de versiones *Subversion* [13]
- Servidor de aplicaciones *Apache Tomcat* [14]
- Servidor de bases de datos *Mysql* [15]
- Servidor de gestión de incidencias *Atlassian Jira* [16]

1.5. Estructura de la memoria

Para facilitar la lectura de la memoria, se incluye a continuación un breve resumen de cada capítulo:

1. Introducción

En este capítulo, encontramos un resumen de la motivación y objetivos de la realización de este proyecto. También se incluye una enumeración de los medios utilizados así como una breve descripción de la estructura del resto del documento.

2. Estado del arte

En este capítulo, se presenta la situación de la empresa que solicita el desarrollo, las soluciones existentes que se encuentran en uso, el estado de las tecnologías de desarrollo de software y se describen las metodologías y tecnologías que se han evaluado hasta seleccionar las que se utilizaron finalmente.

3. Análisis

En este capítulo, se detalla la labor de análisis previa a la implementación: tanto los requisitos funcionales que se desea cubrir como los diagramas de casos de uso de la aplicación.

4. Diseño

En este capítulo, se explica la arquitectura de software elegida para esta aplicación, y se detalla cada uno de los componentes de esta arquitectura: se presenta el modelo de datos y el diseño de la interfaz de usuario.

5. Planificación y presupuesto

En este capítulo, se describe la planificación inicial del proyecto, el diagrama de gantt y el presupuesto junto con el desglose de costes de personal, materiales y totales del desarrollo.

6. Implantación y manual de usuario

En este capítulo, se recoge paso a paso el proceso de instalación del software necesario para el despliegue del proyecto en un servidor de aplicaciones. También se recoge el manual de usuario.

7. Conclusiones y líneas futuras

En este capítulo, se enumeran las conclusiones tanto personales como profesionales obtenidas de la realización de este proyecto, los objetivos alcanzados, y las líneas futuras de desarrollo que quedan pendientes de implementación.

Capítulo 2

Estado del Arte

En este capítulo, se profundiza en la historia de la empresa que solicita el desarrollo, se enumeran las herramientas de gestión utilizadas por la empresa, el estado de las tecnologías de desarrollo de software y se evalúan las metodologías y tecnologías a utilizar en este proyecto.

2.1. Salenda

Salenda [1], es una empresa tecnológica española que ofrece servicios de desarrollo de software, formación IT y gestión de infraestructuras informáticas y telemáticas.

Inicia su andadura en el año 2005, como empresa unipersonal formada por el Arquitecto de Software y emprendedor *Álvaro Sanchez-Mariscal* [17], antiguo alumno de la *Universidad Carlos III de Madrid* [18] (UC3M) y posteriormente miembro del Departamento de Desarrollo de la UC3M, que ha colaborado estrechamente con la universidad durante toda su carrera profesional. Desde 2002 colaboró activamente con el *GUL* [19]. Fue ponente durante años en sus Jornadas Técnicas, organizadas en Marzo y Noviembre cada año, así como miembro de la Junta Directiva de la asociación, en calidad de vocal.

En 2007, Salenda comienza a colaborar con los *CAU* [20] de los campus de Getafe y Leganés de la UC3M contratando Técnicos Informáticos para estos departamentos. Es aquí donde se inicia la relación entre la empresa y el alumno, que posteriormente dará como resultado la realización de este proyecto. En 2009, Salenda crea junto con *ImaginaWorks* [21] una empresa filial de ambas: *Escuela de Groovy* [22]. Se trata de la primera empresa española y latinoamericana enfocada en llevar las tecnologías Groovy y Grails a las empresas.

En este escenario surge la posibilidad de una colaboración entre el alumno y técnico informático del CAU de Leganés *Roberto Martín Flores* [23] (autor de la memoria objeto de este PFC) y la empresa Salenda, quien en su búsqueda por una aplicación que cubra sus necesidades de gestión y seguimiento de proyectos de consultoría, decide desarrollar su propia solución, dado que las herramientas que están utilizando en ese momento tienen un coste de licencia elevado y no cubren completamente todas las funcionalidades requeridas.

2.2. Soluciones existentes

En el momento de la solicitud de realización de este proyecto, Salenda, utiliza tres herramientas de software diferentes para sus principales necesidades de gestión. Cada una de estas soluciones cubre un conjunto de funcionalidades, tiene unos requisitos de hardware específicos y unos costes por licencia individuales, que la empresa soporta.

2.2.1. Jira

Jira [16] es una herramienta que permite el seguimiento de errores e incidencias y la gestión de proyectos.

Esta herramienta fue desarrollada en el año 2004, por la empresa australiana Atlassian, como una aplicación web basada en las tecnologías *J2EE* [24]. Inicialmente Jira se utilizó para el desarrollo de software, sirviendo de apoyo para la gestión de requisitos, seguimiento del estatus y más tarde para el seguimiento de errores, permitiendo mejorar la calidad del código y aumentar la velocidad de desarrollo. Jira también se utiliza en áreas no técnicas para la administración de tareas.

Actualmente Jira se utiliza en 107 países para el seguimiento de incidencias y la gestión de proyectos y cuenta con más de 11.500 clientes, tanto empresariales (*SAP*, *IBM*, *BMW*), como institucionales (*CERN*, *BBC*) y universidades de renombre como *Harvard* o *Stanford* entre muchos otros. Entre las funcionalidades que cubre Jira encontramos:

- Seguimiento de incidencias.

Las incidencias son una parte inevitable del desarrollo de software para todos los equipos de desarrollo (ya sean grandes o pequeños). Jira permite un seguimiento de incidencias fácil, para que el equipo concentre su esfuerzo en lo más importante: crear un buen producto. Con Jira podemos hacer seguimiento de bugs, tareas técnicas o funcionalidades completas, podemos asignarlas a distintos usuarios, conocer su estado, los tiempos invertidos y su resolución.

- Seguimiento de proyectos

Jira permite no solo seguir incidencias de un proyecto, sino de tantos como haya en desarrollo dentro de la empresa, y para cada uno permite distinta planificación, estimación e imputación de horas individuales por usuario, incluso, flujos de trabajo diferentes para cada proyecto. Dado que Jira puede utilizarse para proyectos que no sean de desarrollo de software, también se pueden estimar e imputar horas a proyectos de otras índoles (comerciales, infraestructura, consultoría, formación, etc..)

- Gestión de desarrollo de software

Jira permite, mediante el uso de *plugins*, extender sus funcionalidades cubriendo otras necesidades del desarrollo de software, como la integración con repositorios de código, la gestión de proyectos ágiles de manera visual o la gestión de versiones y calendario de *releases* [25] de una aplicación.

Jira está disponible en dos modalidades de instalación: *On Demand* (Atlassian aloja la herramienta por ti y se encarga de la instalación y el mantenimiento) u *On Site* (El usuario decide dónde instala y mantiene su propia instancia de Jira). Cada modalidad, tiene unos servicios asociados y unos costes unitarios de licencia.

Particularmente, Salenda, había optado por mantener su propia instalación de Jira, alojada en un servidor propio con sistema operativo Linux. Los requisitos de hardware para esta instancia eran:

Como mínimo, un servidor reciente (con CPU multi-núcleo) con 2GB de memoria RAM disponible y un disco duro razonablemente rápido (7200rpm o más rápido).

Esta modalidad de Jira, ofrece 12 meses de mantenimiento, actualizaciones y soporte para hasta 25 usuarios, a un precio de 1800\$ anuales.

2.2.2. SugarCRM

SugarCrm [26] es un sistema para la administración de la *relación con los clientes* [27].

Este sistema fue desarrollado en 2004, por la empresa estadounidense SugarCrm, Inc ubicada en Cupertino (California), basado en las tecnologías *LAMP* [28] (Linux-Apache-MySQL-PHP). Está diseñado para facilitar la gestión de ventas, oportunidades, contactos de negocios y más.

En la actualidad cuenta con más de 1.5 millones de instalaciones en 120 países y está traducido a 26 idiomas diferentes. Entre sus clientes más importantes se encuentran empresas internacionales como *HoneyWell*, *Yahoo* o *Starbucks*, y también instituciones como *El estado de Oregón* o el *Centro de investigación Ames de La Nasa*. Entre las funcionalidades que cubre SugarCRM encontramos:

- Ventas y Marketing

Esta es la funcionalidad más potente de SugarCRM. Permite el seguimiento y especialización de cada contacto de la empresa, desde que es una *oportunidad* de negocio hasta convertirse en una *cuenta*. Permite conocer la última vez que se contactó a un cliente, sus características: teléfono, correo electrónico, reuniones realizadas, contactos realizados, etc. Permite programar próximas llamadas, visitas y reuniones, facilitando la gestión al equipo de ventas.

- Gestión de Proyectos

SugarCRM ofrece la posibilidad, de agrupar uno o varios clientes o acciones comerciales dentro de un *proyecto*. Permite realizar seguimientos, obtener estadísticas y determinar el ROI [29] de una determinada *campana* o actividad comercial, en base a su presupuesto.

- Soporte a cliente

Permite centralizar los productos que tiene contratados un cliente, así como el historial de contactos y comunicaciones que éste ha realizado con el servicio de soporte, permitiendo dar en cada momento el mejor servicio con independencia del técnico que atienda la petición al tener compartida siempre toda la información.

- Integración de las comunicaciones

Permite una visibilidad inmediata y cronológica de todas las comunicaciones establecidas con el cliente vía email, twitter, llamada telefónica o documental. SugarCRM puede integrarse con Gmail [30], Twitter [31], plataformas de vo-ip como GoToMeeting [32], y repositorios documentales como Google Docs [33].

SugarCRM está disponible en dos modalidades de instalación: *On Demand* (SugarCRM inc. aloja la herramienta por ti y se encarga de la instalación y el mantenimiento) u *On Site* (El usuario decide dónde instala y mantiene su propia instancia de SugarCRM). Ambas modalidades, tienen los mismos servicios asociados y costes unitarios de licencia.

Por este motivo, Salenda, había optado por alojar su instancia de SugarCRM en la modalidad *On-Demand* y ahorrarse los costes de un servidor propio.

La modalidad más básica de SugarCRM, ofrece 12 meses de alojamiento, mantenimiento, actualizaciones y soporte para hasta 10 usuarios, a un precio de 4800\$ anuales.

2.2.3. SAGE. ContaPlus, FacturaPlus, NominaPlus

Sage [34] es una empresa que se dedica al desarrollo de soluciones software y a los servicios de gestión empresarial para ayudar a millones de clientes en todo el mundo a ser más competitivos en un mercado global.

Desarrollan soluciones de software para controlar toda la gestión administrativa de una pyme [2], algo fundamental en el ejercicio de cualquier actividad económica: Realizar asientos contables, elaborar nóminas, emitir facturas, revisar la tesorería y/o elaborar presupuestos, son entre otros las muchas soluciones del software de Sage.

Sage está presente en 23 países, cuenta con más de 350.000 clientes y cerca de 2.600 Business Partners. Aunque Sage se fundó en Newcastle hace 30 años, está presente en España desde 2003, anteriormente se conocía como Grupo SP. Entre las soluciones de software que ofrece Sage, las utilizadas por Salenda son:

- ContaPlus

ContaPlus es una herramienta de gestión que permite registrar de manera contable y automática operaciones de compra y venta, conocer la evolución prevista de la tesorería y el flujo de caja, presentar los modelos de IVA e IRPF, y controlar la amortización de los inmovilizados, entre otros.

- FacturaPlus

FacturaPlus es un programa de facturación diseñado para facilitar el control de los procesos comerciales de un negocio: desde la emisión de documentos de compra/venta (facturas) hasta el control del histórico y el estado de tu negocio.

- NominaPlus

NominaPlus es una herramienta de gestión de recursos humanos que permite, controlar salarios y beneficios, elaborar nóminas, así como emitir los gastos de Seguridad Social e IRPF de los trabajadores, entre otros.

Cada una de estas soluciones forma parte de una *Suite de software* que se instala en un Ordenador Personal (o PC). El manual de usuario de la Suite, recomienda los siguientes requisitos mínimos para el correcto funcionamiento de las aplicaciones:

Como mínimo, un pc con procesador Intel Core 2 o superior, 1Gb de memoria RAM, 300Mb de HD disponible, un lector de DVD, conexión a internet para la validación de la licencia y actualizaciones automáticas, Internet Explorer 7.0 o superior, Microsoft Windows XP SP3 así como un lector de documentos PDF.

Salenda utilizaba las 3 aplicaciones anteriores de las disponibles en la Suite, por las que pagaba unos costes de licencia anuales de 900€ en concepto de licencia, mantenimiento, actualizaciones y soporte para 1 única instalación.

2.2.4. Comparativa

Observando la siguiente tabla comparativa [2.1](#) podemos ver cómo PGS cubrirá prácticamente el 90 % de las funcionalidades que ofrecen actualmente los otros 3 softwares en uso.

Algunas de ellas se mantendrán en uso por cuestiones obvias, inicialmente no sería rentable intentar reproducir toda la funcionalidad, y aunque uno de los objetivos de PGS era cubrir ciertas funcionalidades del software actualmente en uso, su principal objetivo es servir como sintetizador de la información de las 3 suites de software actualmente en uso para ofrecer información relevante que permita obtener una ventaja competitiva como *el coste/hora de un empleado o los costes/beneficios de un proyecto en tiempo real en base a las horas y gastos invertidos en el.*

<i>Funcionalidad</i>	Jira	SugarCRM	SAGE	PGS
Gestión de incidencias	✓			
Asignación de incidencias	✓			✓
Horas dedicadas por proyecto / persona	✓			✓
Coste hora empleado				✓
Trabajador más / menos activo				✓
Planificación de proyectos	✓	✓		✓
Presupuestación de Proyectos		✓		✓
Proyectos presupuestado vs cobrado				✓
Proyectos beneficio vs coste				✓
Tiempo invertido por proyecto	✓			✓
ROI acciones comerciales		✓		✓
Seguimiento de clientes		✓		✓
Seguimiento de proveedores				✓
Planificación de llamadas, citas, etc.		✓		
Integración con sistemas de comunic.		✓		
Gestor documental	✓	✓	✓	✓
Gestión de compras			✓	✓
Asignación de compras (amortiz. por proyecto / persona / empr.)				✓
Amortización del inmovilizado			✓	✓
Gestión de nóminas			✓	✓
Gestión de facturas			✓	✓
Balance de caja			✓	✓
Gestión de impuestos (IVA, SS, IRPF)			✓	✓

Tabla 2.1: Soluciones Existentes: Comparativa

2.3. Tecnologías de desarrollo de Software

En el momento de la solicitud de realización de este proyecto, encontramos un panorama excitante en las tecnologías de desarrollo de software. Se trata de una época de cambios, en los que las metodologías ágiles están influyendo intensamente en la manera de codificar y elaborar software. Movimientos como el *Software Craftmanship* [35], *Clean Code* [36] o *TDD* [37] toman relevancia y reconocimiento por toda la comunidad de desarrollo de software, dando lugar a la aparición de nuevos frameworks que permitan de manera sencilla aplicar estas técnicas en el día a día del desarrollo.

2.3.1. Herramientas estandarizadas. Java + J2EE

A finales de la primera década del siglo XXI, J2EE [24] es uno de los frameworks más extendidos en el desarrollo de software de empresas e instituciones públicas de todo el mundo. Este framework fue construido sobre el lenguaje de programación *Java* [38], por la empresa estadounidense *Sun Microsystems* [39] (que posteriormente sería adquirida por su actual propietario *Oracle* [40]).

Ofrece un amplio conjunto de herramientas para el desarrollo de software empresarial, concentrando en una sola plataforma todo lo necesario para dar solución a cualquier problema que podamos encontrar en el flujo de desarrollo de una aplicación: *EJB* [41], *JMS* [42], *Java Servlets* [43], *JSF* [44], ... Además, todas estas herramientas se ofrecen de manera gratuita ya que son tecnologías *Open Source* [45] (de código abierto). J2EE trajo la estandarización y la simplificación al software empresarial, permitiendo elaborar aplicaciones distribuidas en N capas, basadas en componentes modulares ejecutados en un servidor de aplicaciones. Dado que se ejecutan en la JVM [6] estas aplicaciones pueden ser multiplataforma, pueden integrarse con cualquier recurso empresarial existente, cuentan con un modelo de seguridad de protección de datos y de conectores al *API* [46] de *JDBC* [47] para poder acceder al repositorio de datos, independientemente del motor de base de datos en que se ejecute.

Todas estas características propiciaron que multitud de organismos públicos, grandes empresas privadas y pymes [2] de todo el mundo apostaran por esta plataforma, organizando toda su infraestructura y modelo de trabajo entorno a ella. La aplicación de este

framework ha permitido durante mucho tiempo, resolver algunos de los problemas más comunes en proyectos a medida para clientes privados e incluso en proyectos Open Source, y esto ha propiciado su expansión y estandarización, dejando un margen muy pequeño a las nuevas tecnologías y frameworks de desarrollo en el entorno empresarial. Sin embargo, algo que en principio parece muy beneficioso, en la práctica resulta complejo:

- Definir la configuración de la aplicación requiere de la elaboración de complejos ficheros xml, lo que supone tiempo y esfuerzo.
- Poner en marcha un nuevo proyecto, requiere de la configuración de los entornos de desarrollo, pruebas y producción, para lo cual se emplean demasiadas horas y dedicación.
- Encontrar la raíz de un problema durante el proceso de desarrollo nos puede llevar varias horas.
- A veces, entroncar un error en el código se vuelve una tarea tediosa e inalcanzable.

Todo esto va en detrimento de la productividad del desarrollador y la satisfacción del cliente. El desarrollador se encuentra demasiados obstáculos que le alejan de su objetivo real, desarrollar la lógica de negocio de la aplicación, y por tanto su motivación y productividad se ven mermadas. A su vez, los plazos de entrega se alargan, y la calidad de las entregas pueden no cumplir las expectativas del cliente, lo cual disminuye su satisfacción y confianza.

Todo esto, motivó la aparición de pensamientos y corrientes que demandaban que el desarrollo de software fuera una “tarea más sencilla”, e incluso llegar a “programar como pensamos”. Esto propició la aparición de los lenguajes dinámicos y frameworks ágiles.

2.3.2. Nuevas tecnologías. Groovy + Grails

La comunidad de desarrollo de software, continua buscando nuevas soluciones que complementen y mejoren el entorno Java, volviendolo más flexible y dinámico, y buscando soluciones a los problemas anteriormente citados. Una de estas soluciones es la ejecución de *lenguajes dinámicos* [48] en la JVM enriqueciendo los recursos de los que ya se disponen.

Los lenguajes dinámicos, permiten crear y modificar tipos y estructuras de datos en tiempo de ejecución, mientras que los denominados lenguajes *estáticos* [49] (como Java) requieren declarar todos los tipos de datos mediante código fuente y precisan de un compilador para una vez traducidos poder ejecutarlos. El principal objetivo de los lenguajes dinámicos, es ofrecer al programador una manera más breve, sencilla, expresiva y fácil de generar y mantener el código de una aplicación.

Groovy [7] es un lenguaje dinámico hecho en Java y para Java. Groovy nos aporta todas las ventajas de un lenguaje dinámico, más la robustez de un lenguaje solido y tipado como Java, nos permite acceder al api del lenguaje y además nos ofrece otras funcionalidades como:

- sobrecarga de operadores
- declaración y creación de objetos en tiempo de ejecución
- uso de closures [50] (similares a las Lambdas [51] de Java8)
- manejo de colecciones de manera sencilla

Cualquier código que podamos hacer con Java, podemos hacerlo con Groovy, porque Groovy es Java en su interior, y por lo tanto también podemos ejecutarlo en la JVM y utilizarlo en conjunto con sus frameworks.

En el momento de realización de este proyecto, *Grails* [5] es el framework de desarrollo más popular basado en Groovy y construido sobre la plataforma J2EE. Grails fue desarrollado por la empresa G2One [52], propiedad de la compañía SpringSource [53] (el equipo detrás del renombrado *Spring Framework* [54]) y ha sido un caso de éxito en muchas compañías y proyectos tales como Netflix, SkyTV, BestBuy, LinkedIn, Vodafone o MTV entre otros.

Al ser un framework basado en J2EE permite a las empresas migrar sus entornos de desarrollo de Java a Grails fácilmente. Cualquier proyecto Grails se puede desplegar en los mismos servidores de aplicaciones que un proyecto J2EE, incluso se pueden utilizar las mismas herramientas de construcción y librerías que se estuvieran utilizando para proyectos Java.

La curva de aprendizaje del lenguaje Groovy y su framework Grails es baja. Cualquier programador con algo de experiencia en Java necesitará muy poco tiempo para comenzar a trabajar con soltura, incluso, podría empezar a utilizar el framework escribiendo su código en Java, e ir migrando poco a poco hacia las nuevas funcionalidades que ofrece Groovy. El otro factor que permite este rápido aprendizaje es que Grails se basa en el paradigma “Convention over Configuration” [8] (“Convención en lugar de configuración”) lo que significa que en Grails apenas necesitamos escribir código xml para configurar *Spring* o *Hibernate* [55]

2.3.3. Java vs Groovy

Veamos algunos ejemplos de cómo el código groovy simplifica el código java. Observe-mos el siguiente fragmento de código, que contiene un problema recurrente en cualquier aplicación, leer un fichero de disco llamado *file.txt*, este sería el código escrito en java:

```
BufferedReader br = new BufferedReader(new FileReader("file.txt"));
try {
    StringBuilder sb = new StringBuilder();
    String line = br.readLine();

    while (line != null) {
        sb.append(line);
        sb.append(System.lineSeparator());
        line = br.readLine();
    }
    String everything = sb.toString();
} finally {
    br.close();
}
```

Figura 2.1: FileReader.java

Este sería el mismo fragmento de código escrito en groovy:

```
File file = new File("file.txt")
String everything = file.text
```

Figura 2.2: FileReader.groovy

En el siguiente ejemplo, vemos todas las “facilidades” que nos aporta este lenguaje dinámico frente a los lenguajes estáticos:

```
public class Saludador {  
    private String usuario;  
  
    public String getUsuario() {  
        return usuario;  
    }  
  
    public void setUsuario(usuario) {  
        this.usuario = usuario;  
    }  
  
    public String saluda(String nombre){  
        return ("Hola " + nombre + ", yo soy " + usuario);  
    }  
}  
  
Saludador saludador = new Saludador();  
saludador.setUsuario("Pablo");  
System.out.println(saludador.saluda("Guillaume"));
```

Figura 2.3: Saludador.java

- Puntos y coma opcionales
- Returns opcionales (última línea valor retornado)
- Privacidad de la clase opcional (todo es Public por defecto)
- Declaración de tipos opcionales (def tipo indefinido)
- Atajos y macros como *println* (equivale a `System.out.println` de java)
- Constructor por defecto (acepta un mapa y hace binding auto.)
- Getters y Setters por defecto (no hace falta crearlos)
- Construcción dinámica de cadenas (GStrings)

Si, aplicamos todos estos beneficios al código anterior, así quedaría el mismo código escrito en groovy:

```
class Saludador {  
    private String usuario  
  
    String saluda(String nombre){  
        "Hola $nombre, yo soy $usuario"  
    }  
}  
  
def saludador = new Saludador(usuario:"Pablo")  
println saludador.saluda("Guillaume")
```

Figura 2.4: Saludador.groovy

Podemos observar que groovy, realmente, simplifica la labor del programador.

2.4. Metodologías de desarrollo de Software

Uno de los objetivos de este proyecto, es llevar a cabo un desarrollo de calidad, que cumpla las expectativas de la empresa que lo solicita, optimizando los costes y el tiempo de desarrollo de manera que se obtenga el mayor valor posible con la menor inversión.

Para poder alcanzar este objetivo, necesitamos un proceso de desarrollo que nos aporte las pautas a seguir, defina los hitos intermedios que se deben alcanzar hasta completar la implementación y establezca los puntos de control necesarios para verificar el valor de cada entrega y tomar las decisiones correctoras oportunas con suficiente antelación. Todas estas pautas podemos encontrarlas en una *Metodología de desarrollo*.

2.4.1. Evolución histórica

En Ingeniería del Software, se define una *Metodología de desarrollo de software* (o *Ciclo de vida del software*) como un conjunto de normas para dividir la labor de implementación del software en distintas fases y acciones con la intención de mejorar la planificación y gestión del proceso de desarrollo. A menudo se las considera un subconjunto del *ciclo de vida de desarrollo de sistemas*. Una metodología, puede incluir la definición de entregables

o artefactos específicos que deben ser elaborados y completados por un equipo de desarrollo para evolucionar o mantener una aplicación.

A lo largo de la historia del desarrollo de software nos encontramos con multitud de modelos de ciclo de vida de software o metodologías de desarrollo, como el modelo en *cascada* (o waterfall), el prototipado (o prototype), el iterativo-incremental, el espiral, o las más recientes como extreme programming (o XP) y las metodologías ágiles como Scrum o Kanban.

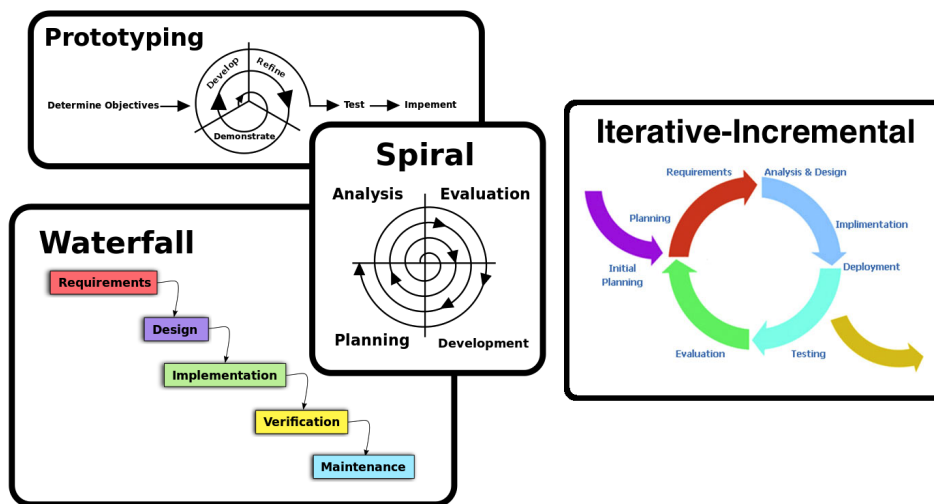


Figura 2.5: Ciclo de vida del software

Cascada vs Iterativo-Incremental

Una de las principales diferencias entre los modelos más conservadores como el de cascada (waterfall) y los evolutivos como el iterativo-incremental, es que el ciclo de pruebas se lleva a cabo en diferentes etapas del ciclo de vida de desarrollo del software. En el modelo en cascada, siempre hay una fase de pruebas independiente, tras la finalización de la fase de implementación. Sin embargo, en los modelos iterativos, se suele hacer o bien de manera concurrente (TDD) o al menos en la misma iteración que la implementación.

Dado que cada pequeña iteración, contiene una fase de pruebas - tras la cual se libera un pequeño entregable - los clientes pueden probar de manera frecuente estas pequeñas piezas de software y validar el valor del producto entregado. Este entrega continua de valor, permite al cliente conocer la evolución de su desarrollo y así, poder tomar mejores

decisiones sobre el futuro de su producto. Tener una retrospectiva del valor entregado, y una sesión de replanificación del proyecto tras cada iteración - Scrum especifica una duración máxima de 1 mes para cada iteración - ayuda al equipo a adaptar su planificación continuamente para maximizar el valor de las entregas.

Todas estas prácticas iterativas, introducen un cambio en la visión del desarrollo hacia una mentalidad “*orientada a producto*” frente a la mentalidad del modelo en cascada más “*orientada a proyecto*”.

Metodologías Ágiles

Las “*Metodologías Ágiles de Desarrollo*” se refieren a un conjunto de metodologías de desarrollo basadas en el modelo Iterativo-Incremental, en las que los requisitos y los entregables, surgen de la colaboración entre equipos auto-organizados y multi-funcionales. El término fue acuñado en el año 2001 con la publicación del Manifiesto Ágil [56].

El *manifiesto ágil*, fue el resultado de una reunión de 17 de los más relevantes desarrolladores de software en el año 2001, en la que, tras discutir sobre las metodologías ágiles de desarrollo publicaron un documento, en el que expresaban las siguientes ideas y se comprometían con ellas:

“Estamos descubriendo formas mejores de desarrollar software tanto por nuestra propia experiencia como ayudando a terceros. A través de este trabajo hemos aprendido a valorar:”

*“**Individuos e interacciones** sobre procesos y herramientas”*

*“**Software funcionando** sobre documentación extensiva”*

*“**Colaboración con el cliente** sobre negociación contractual”*

*“**Respuesta ante el cambio** sobre seguir un plan”*

Esto significaba que, basándose en sus experiencias como desarrolladores de software, aunque valoraban los elementos de la derecha, valoraban más los de la izquierda. El significado de los elementos de la izquierda del manifiesto es:

- **Los individuos y las interacciones:** la auto-organización y la motivación son

importantes, como lo son las interacciones como co-ubicación y la programación en parejas.

- **Software funcionando:** las entregas mínimas de software que funciona son más útiles y agradecidas que las presentaciones y documentos en las reuniones con los clientes.
- **Colaboración con el cliente:** en la mayoría de los casos, los requisitos no pueden ser recogidos completamente al principio del ciclo de desarrollo de software, por lo tanto, la implicación y participación continua del cliente es muy importante.
- **Respuesta al cambio:** los métodos ágiles, se centran en la rápida respuesta al cambio y a la integración continua del desarrollo.

En el momento de realización de este proyecto, Scrum es la metodología más utilizada en aquellas empresas españolas, que se encuentran inmersas en un cambio hacia una gestión de procesos ágil (como es el caso de Salenda).

2.4.2. Scrum

Scrum es un marco de trabajo (o framework) para el desarrollo ágil de productos, no teniendo por qué ser de software, aunque aquí trataremos su versión aplicada al software. Tiene su origen en el año 1995 y fue co-desarrollada por Ken Schwaber y Jeff Sutherland, basándose en el modelo de desarrollo identificado por Nonaka & Takeuchi en 1986 que se utilizaba en las principales empresas de manufactura tecnológica de la época. El nombre tiene su origen de la comparación de esta nueva forma de trabajo en equipo, con el avance en formación de melé (scrum en inglés) de los jugadores de Rugby. Se basa en una serie de principios, prácticas y valores:

- Entrega temprana al cliente (de 2 semanas a 2 meses) de software con valor para su satisfacción.
- No alentar la resistencia al cambio, como es la tónica habitual en la gestión orientada a proyectos, en su lugar, fomentar la flexibilidad para aumentar la ventaja competitiva del cliente y su satisfacción.

- Entregas iterativas e incrementales, aportando nueva funcionalidad en cada iteración (o *sprint*), intentando alcanzar un ritmo constante de desarrollo.

Se fomenta el trabajo en equipos auto-organizados, en los que la colaboración y comunicación entre profesionales prima sobre las jerarquías y la cadena de mando. En lugar de puestos de dirección o divisiones jerárquicas, scrum, identifica una serie de roles [57], reuniones y artefactos, necesarios para la consecución del desarrollo.

Artefactos

Sprint es el nombre que recibe cada iteración en la que se realiza el desarrollo en sí. Se recomienda que la duración sea inicialmente de 2 semanas, pero constante y definida por el equipo en base a su experiencia, quien irá ajustándola según su propio ritmo, sin relajarlo demasiado. Al final de cada sprint, el equipo deberá presentar los avances logrados, y el resultado obtenido debe ser un producto potencialmente entregable. Así mismo, se recomienda no agregar objetivos al sprint, a menos, que la falta de estos objetivos amenace el éxito del proyecto.

Product Backlog es un conjunto “abierto” de todos los requisitos de proyecto. Contiene descripciones genéricas de funcionalidades deseables, priorizadas según su retorno de inversión [29] que solo puede ser modificado por el dueño del producto. Cada requisito se acompaña de estimaciones realizadas a grandes rasgos, tanto del valor para el negocio, como del esfuerzo requerido para su implementación, de manera que en cada iteración, puede ser re-priorizado y modificado para maximizar el roi y la satisfacción del cliente.

Sprint Backlog es el subconjunto de requisitos que serán desarrollados durante el siguiente sprint. Al definir el *sprint backlog* el equipo describe cómo va a implementar los requisitos, dividiéndolos en tareas y estimando a grandes rasgos el esfuerzo que requerirá la consecución de cada una. Estas tareas, nunca son pre asignadas, en su lugar cada miembro del equipo tomará la siguiente por orden de prioridad según priorizó el dueño del producto anteriormente.

Roles

Product Owner es la figura que representa los intereses del cliente, escribe las *historias de usuario* (requisitos de negocio) y las prioriza en el *Product Backlog*

Scrum Master es la figura que vela por el proceso y facilita su ejecución eliminando los obstáculos que impiden al equipo alcanzar el objetivo del *Sprint*. No es el líder del equipo, sino que actúa como su protector ante cualquier influencia externa que les distraiga.

Equipo de desarrollo son quienes tienen la responsabilidad real de entregar el desarrollo. Se recomienda un tamaño de equipo entre 3 y 9 personas con habilidades transversales (análisis, diseño, desarrollo, pruebas, documentación, etc...)

Stakeholders este rol representa a los clientes, inversores, vendedores, la gente que solicita el proyecto, y para quienes producirá un beneficio real. Sólo participan durante las revisiones de cada *sprint*.

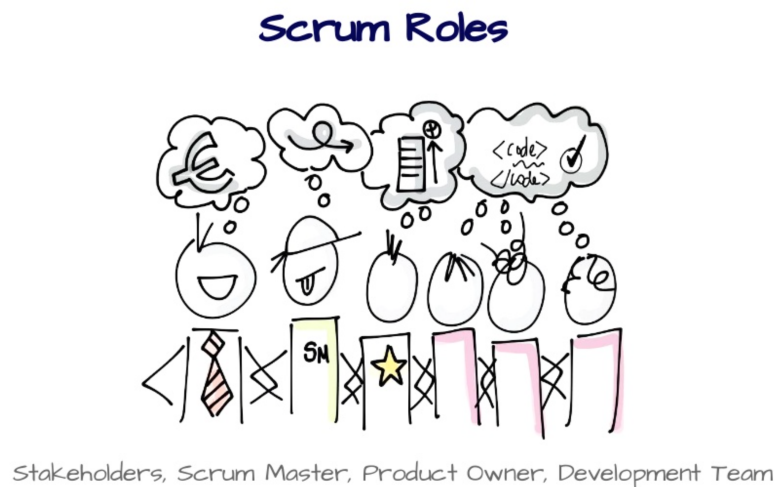


Figura 2.6: Scrum Roles

Reuniones

Product Backlog Grooming en esta reunión el *Product Owner*(PO) presenta al *equipo* y al *Scrum Master*(SM) los requisitos del proyecto, se plantean todas las dudas respecto a las especificaciones y se realiza una primera estimación general y priorización de los elementos del *product backlog*.

Sprint Planning en esta reunión vuelven a participar los roles anteriores, pero esta vez, para decidir y planificar en detalle qué parte del *product backlog* se llevará a cabo durante este *sprint* (lo que compondrá el *sprint backlog*) y cómo van a hacerlo exactamente. Al final de esta reunión el PO priorizará las tareas que el equipo se ha comprometido a llevar a cabo durante el *sprint* según su roi y esfuerzo estimado.

Daily Scrum o daily stand-up meeting es una reunión diaria en la que el SM y el equipo analizan brevemente el estado del sprint en curso. Se realiza puntualmente todos los días a la misma hora y no debe superar los 15 minutos de duración. Durante la reunión cada miembro da respuesta a 3 preguntas: - ¿Qué hiciste ayer?, - ¿Qué vas a hacer hoy?, - ¿Tienes algún problema que te impida alcanzar tu objetivo? (Será responsabilidad del SM solventar estos impedimentos para que el equipo pueda continuar su labor)

Sprint Review es una reunión en la que *el equipo* presenta al PO los objetivos alcanzados en el sprint (el incremento) y el SM revisa los items del Backlog que se han completado y los que no (analizando con el equipo por qué no se han podido completar). Finalmente el PO puede reorganizar el Backlog del proyecto si lo considera necesario en vista del resultado del sprint.

Sprint Retrospective es una reunión en la que el equipo y el SM analizan cómo ha ido el sprint, y cómo mejorarlo persiguiendo siempre la *mejora continua* del proceso y el equipo.

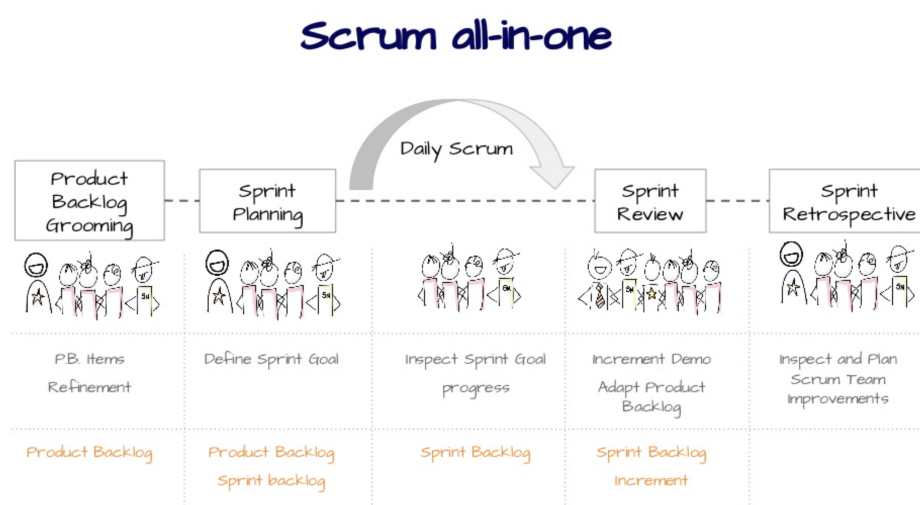


Figura 2.7: Scrum Meetings

Después de tomar contacto con la metodología, y de recibir una breve formación por parte de la empresa (Salenda), se decide apostar por ésta para la consecución de este proyecto. Se plantea por tanto la siguiente hoja de ruta:

1. Se llevará a cabo una primera reunión de análisis entre el PO que nombre la empresa y el desarrollador para identificar: los principales objetivos y requisitos funcionales, así como los actores y casos de uso que tomarán partido en la aplicación.
2. Se elaborará un Product Backlog inicial que contendrá los requisitos identificados en la reunión anterior.
3. El desarrollador elaborará mockups (bocetos) del diseño de cada parte de la aplicación, que deberán ser aprobados antes de cada inicio de iteración por el PO.
4. Se planificarán sprints de 2 a 4 semanas de duración compuestos de las siguientes fases:
 - día1 - Sprint planning, el PO y el desarrollador abordarán en detalle las tareas a completar en las próximas 2/4 semanas.
 - día2 - Mockups, se elaborarán y aprobarán los bocetos de la siguiente parte de la aplicación que se vaya a abordar. (Si alguna iteración no requiere de bocetos, se iniciará la siguiente fase directamente)
 - días 3 a 9 - Implementación y pruebas de las tareas del sprint. Diariamente el desarrollador se reunirá con el SM designado por la empresa para comentarle cualquier impedimento que este pudiendo tener.
 - día 10 - Sprint review, se presentará al PO el resultado de las tareas abordadas, se tomarán nuevos requisitos de ser necesario y se añadirán al product backlog. Finalmente se desplegará el entregable obtenido en un servidor de pruebas de la empresa para su verificación a lo largo del próximo sprint.
 - Durante el transcurso de un sprint, la empresa podrá priorizar el product backlog como mejor le convenga. Sin embargo, se intentará no modificar el alcance del sprint, a no ser, que de no hacerlo se pusiera en peligro la consecución del proyecto.

2.5. Prueba de concepto

A la vista de estas nuevas tecnologías y metodologías de desarrollo, parece que nos encontramos ante las herramientas idóneas para alcanzar nuestros objetivos: “*desarrollar un software de calidad minimizando el tiempo de desarrollo y sus costes*”. Sin embargo para estar seguros, debemos realizar alguna prueba empírica de la aplicación de estas tecnologías.

Para llevar a cabo este POC [58] (Prueba De Concepto) necesitamos en primer lugar un IDE [11] que nos permita desarrollar en el lenguaje y framework elegidos con un mínimo de integración. Algunos de estos como Netbeans [59] o Eclipse [60] aunque son herramientas extensamente utilizadas y maduras, en el momento de realización de esta prueba, no llegaban al nivel de integración requerido, al menos no tanto como el entorno de desarrollo SpringSource Tool Suite [12] (posteriormente renombrado a Groovy/Grails Tool Suite), herramienta desarrollada por la misma empresa responsable del desarrollo del framework Grails.

Descargamos el JDK [61] 1.6 de Java necesario para poder ejecutar los binarios de Grails y configuramos sus variables de entorno. Descargamos y descomprimos los binarios de Grails. Instalamos SpringSource Tool Suite. No necesitamos instalar un servidor de aplicaciones puesto que el framework nos proporciona un servidor *Jetty* [62] embebido. Tampoco requerimos instalar un motor de base de datos, pues el framework nos proporciona una base de datos en memoria *HSQLDB* [63] (aunque podríamos integrar nuestra aplicación con cualquier gestor de base de datos y desplegarla en cualquier servidor de aplicaciones).

Siguiendo la convención de Grails (descrita en su extensa documentación [64]) definimos algunas clases de dominio (*Empleado*, *Proyecto* y *Worklog*) y sus propiedades (solo las suficientes para comprobar alguna funcionalidad básica). Ahora hacemos uso de una de las mejores herramientas del framework, el “scaffolding” [65]. Esta herramienta genera automáticamente *controladores* y *vistas* siguiendo el modelo *MVC* [66] (Modelo - Vista - Controlador) en base a las clases de dominio (*Modelo*) que hemos definido anteriormente, obteniendo una aplicación que nos permite insertar, modificar, listar y borrar instancias del modelo en la base de datos.

Ahora creamos un servicio con un método que reciba como entrada un proyecto, y devuelva como salida las horas trabajadas en el mismo. Haciendo uso de GORM [67] (Grails Object Relational Mapping) y sus Dynamic-Finders, en menos de 10 líneas de código, podemos obtener el sumatorio de horas trabajadas por todos los empleados en el proyecto indicado. Modificando ligeramente las vistas generadas por el scaffolding podemos mostrar el total de horas trabajadas junto a cada proyecto.

Llegados a este punto, en menos de 1 hora hemos podido generar lógica de negocio útil para el proyecto, sin necesidad de tener que configurar interminables ficheros xml, ni declarar multitud de conectores a base de datos o sus DAOs [68].

A la vista de estos resultados, finalmente se toma la decisión de utilizar el framework Grails y la metodología Scrum.

Capítulo 3

Análisis

En este capítulo, se profundiza en la labor de análisis llevada a cabo antes de la fase de implementación. En él se detallan, los requisitos funcionales y no funcionales obtenidos de la primera fase de toma de requisitos y se presentan tanto los actores, como los casos de uso que fueron identificados durante la fase de análisis de los requisitos anteriores.

3.1. Identificación de Requisitos

En este apartado se recogen los *requisitos funcionales*, todas aquellas funciones y comportamientos que el sistema y sus componentes deben cumplir; así como los *requisitos no funcionales*, es decir todos los aspectos no contenidos en los requisitos funcionales que se esperan del sistema, como la disponibilidad, usabilidad o el rendimiento.

3.1.1. Plantilla

Cada requisito será identificado de manera única, y para facilitar su lectura y comprensión, se utilizará la siguiente plantilla reflejada en la tabla 3.1:

<i>Identificador</i>	{identificador}	<i>Título</i>	{título}
<i>Módulo</i>	{módulo}	<i>Prioridad</i>	{prioridad}
<i>Descripción</i>	{descripción}		
<i>Observaciones</i>	{observaciones}		

Tabla 3.1: Plantilla de requisitos

Leyenda:

- **Identificador:** Identificador único del requisito. Compuesto por un prefijo y una clave numérica. El prefijo de esta clave servirá para identificar la naturaleza del requisito, así RF-XX se tratará de un *requisito funcional*. Mientras que RNF-XX se tratará de un *requisito no funcional*.
- **Título:** título descriptivo del requisito.
- **Módulo:** módulo de la aplicación en el que se encuadra el requisito. Puede ser genérico y no pertenecer a un módulo particular.
- **Prioridad:** tomará uno de los siguientes valores alta, media, baja.
- **Descripción:** breve texto explicativo del requisito.
- **Observaciones:** dará información adicional acerca de los requisitos o aspectos internos a tener en cuenta.

3.1.2. Requisitos Funcionales

Un requisito funcional describe una función del sistema o sus componentes, tales como cálculos, manipulación de datos y otros comportamientos que el sistema debe cumplir.

<i>Identificador</i>	RF-1	<i>Título</i>	Módulo de empleados
<i>Módulo</i>	Empleados	<i>Prioridad</i>	Alta
<i>Descripción</i>	Toda la lógica de negocio relativa a los empleados debe estar contenida dentro de una única vista.		
<i>Observaciones</i>	El módulo debe ser seleccionable desde una pestaña (tab).		

Tabla 3.2: RF-1. Módulo de empleados

<i>Identificador</i>	RF-2	<i>Título</i>	Listado de empleados
<i>Módulo</i>	Empleados	<i>Prioridad</i>	Alta
<i>Descripción</i>	El módulo de empleados mostrará un listado, ordenado y paginado de todos los empleados de la empresa.		
<i>Observaciones</i>	El listado se podrá ordenar por cualquier columna. El listado se podrá filtrar por el estado de los empleados: <i>Activos</i> , <i>Inactivos</i> , <i>Todos</i> . El listado mostrará los empleados en estado <i>activo</i> por defecto. El listado mostrará el coste/hora y las horas/mes, de cada empleado, para el mes en curso.		

Tabla 3.3: RF-2. Listado de empleados

<i>Identificador</i>	RF-3	<i>Título</i>	Alta de empleado
<i>Módulo</i>	Empleados	<i>Prioridad</i>	Alta
<i>Descripción</i>	El módulo de empleados tendrá un botón, que abrirá el formulario de creación de un empleado.		
<i>Observaciones</i>	Un empleado será identificado de manera unívoca por su DNI. Al guardar un empleado, el sistema informará de los campos requeridos no completados, y de aquellos que contengan errores.		

Tabla 3.4: RF-3. Alta de empleado

<i>Identificador</i>	RF-4	<i>Título</i>	Modificación de empleado
<i>Módulo</i>	Empleados	<i>Prioridad</i>	Alta
<i>Descripción</i>	El sistema, mostrará el formulario de modificación de empleado, haciendo click en alguno de los campos del listado.		
<i>Observaciones</i>	Cuando no se cumplan los requisitos de modificación, el sistema mantendrá los datos antiguos y los mostrará de nuevo, informando de los errores.		

Tabla 3.5: RF-4. Modificación de empleado

<i>Identificador</i>	RF-5	<i>Título</i>	Baja de empleado
<i>Módulo</i>	Empleados	<i>Prioridad</i>	Alta
<i>Descripción</i>	El sistema, permitirá dar de baja a un empleado, desde el formulario de modificación.		
<i>Observaciones</i>	La baja, no borrará los datos del empleado. El sistema pedirá confirmación para de baja al empleado. Un empleado de baja, se debe poder dar de alta.		

Tabla 3.6: RF-5. Baja de empleado

<i>Identificador</i>	RF-6	<i>Título</i>	Eliminación de empleado
<i>Módulo</i>	Empleados	<i>Prioridad</i>	Baja
<i>Descripción</i>	El sistema, permitirá eliminar definitivamente un empleado dado de baja.		
<i>Observaciones</i>	El sistema pedirá confirmación para eliminar un empleado.		

Tabla 3.7: RF-6. Eliminación de empleado

<i>Identificador</i>	RF-7	<i>Título</i>	Gráfica coste/hora
<i>Módulo</i>	Empleados	<i>Prioridad</i>	Media
<i>Descripción</i>	El sistema, mostrará una gráfica del coste/hora mensual de cada empleado.		
<i>Observaciones</i>	Los datos de la gráfica serán relativos al mes en curso.		

Tabla 3.8: RF-7. Gráfica de coste/hora

<i>Identificador</i>	RF-8	<i>Título</i>	Cálculo del coste/hora
<i>Módulo</i>	Empleados	<i>Prioridad</i>	Alta
<i>Descripción</i>	El sistema, calculará el coste/hora de cada empleado activo.		
<i>Observaciones</i>	El coste/hora, se calculará dividiendo el coste total de la paga de un empleado, entre el número de horas trabajadas, para el mes en curso.		

Tabla 3.9: RF-8. Cálculo del coste/hora

<i>Identificador</i>	RF-9	<i>Título</i>	Gráfica horas/mes
<i>Módulo</i>	Empleados	<i>Prioridad</i>	Media
<i>Descripción</i>	El sistema, mostrará una gráfica de las horas mensuales de cada empleado.		
<i>Observaciones</i>	Los datos de la gráfica serán relativos al mes en curso.		

Tabla 3.10: RF-9. Gráfica de horas/mes

<i>Identificador</i>	RF-10	<i>Título</i>	Cálculo de las horas/mes
<i>Módulo</i>	Empleados	<i>Prioridad</i>	Alta
<i>Descripción</i>	El sistema, calculará las horas/mes trabajadas de cada empleado.		
<i>Observaciones</i>	El número de horas trabajadas por un empleado, se calculará como el sumatorio del tiempo empleado, en todos los worklogs imputados a ese empleado, con fecha de actividad dentro del mes en curso.		

Tabla 3.11: RF-10. Cálculo de las horas/mes

<i>Identificador</i>	RF-11	<i>Título</i>	Listado de últimas pagas empleado
<i>Módulo</i>	Empleados	<i>Prioridad</i>	Alta
<i>Descripción</i>	El formulario de modificación de un empleado, mostrará un listado, ordenado y paginado de las últimas pagas mensuales creadas para el empleado.		
<i>Observaciones</i>	El listado se podrá ordenar por cualquier columna. El listado estará ordenado por mes y año de la paga de manera descendente.		

Tabla 3.12: RF-11. Listado de últimas pagas empleado

<i>Identificador</i>	RF-12	<i>Título</i>	Alta de paga empleado
<i>Módulo</i>	Empleados	<i>Prioridad</i>	Alta
<i>Descripción</i>	El listado de últimas pagas tendrá un botón, que abrirá el formulario de creación de una paga.		
<i>Observaciones</i>	Una paga será identificada de manera unívoca por su mes y su año. El campo <i>Salario Bruto</i> será pre-calculado automáticamente por el sistema como la división del <i>Salario Bruto Anual</i> del empleado entre 12 meses. El campo <i>SS Empresa</i> será pre-calculado automáticamente por el sistema como el 6 % del <i>salario bruto mensual</i> . Ambos campos podrán ser sobreescritos por el usuario. Al guardar una paga, el sistema informará de los campos requeridos no completados, y de aquellos que contengan errores.		

Tabla 3.13: RF-12. Alta de paga empleado

<i>Identificador</i>	RF-13	<i>Título</i>	Modificación de paga empleado
<i>Módulo</i>	Empleados	<i>Prioridad</i>	Alta
<i>Descripción</i>	El sistema, mostrará el formulario de modificación de una paga mensual, haciendo click en alguno de los campos del listado.		
<i>Observaciones</i>	Los campos <i>Coste Asumido</i> y <i>Coste total</i> , no serán editables por el usuario pero se podrán visualizar. Cuando no se cumplan los requisitos de modificación, el sistema mantendrá los datos antiguos y los mostrará de nuevo, informando de los errores.		

Tabla 3.14: RF-13. Modificación de paga empleado

<i>Identificador</i>	RF-14	<i>Título</i>	Eliminación de paga empleado
<i>Módulo</i>	Empleados	<i>Prioridad</i>	Media
<i>Descripción</i>	El sistema, permitirá eliminar definitivamente una paga empleado.		
<i>Observaciones</i>	El sistema pedirá confirmación para eliminar una paga empleado.		

Tabla 3.15: RF-14. Eliminación de paga empleado

<i>Identificador</i>	RF-15	<i>Título</i>	Módulo de compras
<i>Módulo</i>	Compras	<i>Prioridad</i>	Alta
<i>Descripción</i>	Toda la lógica de negocio relativa a las compras y los proveedores debe estar contenida dentro de una única vista.		
<i>Observaciones</i>	El módulo debe ser seleccionable desde una pestaña (tab).		

Tabla 3.16: RF-15. Módulo de compras

<i>Identificador</i>	RF-16	<i>Título</i>	Listado de proveedores
<i>Módulo</i>	Compras	<i>Prioridad</i>	Alta
<i>Descripción</i>	El módulo de compras mostrará un listado, ordenado y paginado de todos los proveedores de la empresa.		
<i>Observaciones</i>	El listado se podrá ordenar por cualquier columna.		

Tabla 3.17: RF-16. Listado de proveedores

<i>Identificador</i>	RF-17	<i>Título</i>	Alta de proveedor
<i>Módulo</i>	Compras	<i>Prioridad</i>	Alta
<i>Descripción</i>	El módulo de compras tendrá un botón, que abrirá el formulario de creación de un proveedor.		
<i>Observaciones</i>	Al guardar un proveedor, el sistema informará de los campos requeridos no completados, y de aquellos que contengan errores.		

Tabla 3.18: RF-17. Alta de proveedor

<i>Identificador</i>	RF-18	<i>Título</i>	Modificación de proveedor
<i>Módulo</i>	Compras	<i>Prioridad</i>	Alta
<i>Descripción</i>	El sistema, mostrará el formulario de modificación de proveedor, haciendo click en alguno de los campos del listado de proveedores.		
<i>Observaciones</i>	Cuando no se cumplan los requisitos de modificación, el sistema mantendrá los datos antiguos y los mostrará de nuevo, informando de los errores.		

Tabla 3.19: RF-18. Modificación de proveedor

<i>Identificador</i>	RF-19	<i>Título</i>	Eliminación de proveedor
<i>Módulo</i>	Compras	<i>Prioridad</i>	Baja
<i>Descripción</i>	El sistema, permitirá eliminar definitivamente un proveedor.		
<i>Observaciones</i>	El sistema pedirá confirmación para eliminar un proveedor. No se podrá eliminar un proveedor en uso.		

Tabla 3.20: RF-19. Eliminación de proveedor

<i>Identificador</i>	RF-20	<i>Título</i>	Listado de compras
<i>Módulo</i>	Compras	<i>Prioridad</i>	Alta
<i>Descripción</i>	El módulo de compras mostrará un listado, ordenado y paginado de todas las compras de la empresa.		
<i>Observaciones</i>	El listado se podrá ordenar por cualquier columna. Junto a cada compra se mostrará el <i>importe</i> total y el porcentaje <i>amortizado</i> acumulado.		

Tabla 3.21: RF-20. Listado de compras

<i>Identificador</i>	RF-21	<i>Título</i>	Alta de compra
<i>Módulo</i>	Compras	<i>Prioridad</i>	Alta
<i>Descripción</i>	El módulo de compras tendrá un botón, que abrirá el formulario de creación de una compra.		
<i>Observaciones</i>	Una compra estará asociada a: <i>la empresa usuario</i> , <i>un proyecto</i> o <i>un empleado</i> ; tendrá (opcionalmente) un <i>tipo de compra</i> que determinará su periodo de amortización (modificable); estará asociada a un proveedor. El formulario de creación de compra permitirá crear un proveedor; éste proveedor será seleccionado automáticamente como el proveedor de la compra. El cliente de una compra siempre será la empresa usuario. La fecha de compra y de pago pueden diferir, pero la fecha de pago no puede ser anterior a la de compra. Cada compra permitirá adjuntar un fichero. Al guardar una compra, el sistema informará de los campos requeridos no completados, y de aquellos que contengan errores. Al guardar una compra se creará una transacción de pago, coincidente con la fecha de pago y el importe de la misma.		

Tabla 3.22: RF-21. Alta de compra

<i>Identificador</i>	RF-22	<i>Título</i>	Modificación de compra
<i>Módulo</i>	Compras	<i>Prioridad</i>	Alta
<i>Descripción</i>	El sistema, mostrará el formulario de modificación de compra, haciendo click en alguno de los campos del listado de compras.		
<i>Observaciones</i>	Una compra será modificable siempre que no se haya iniciado su amortización. Para una compra con amortización iniciada, solo se podrá modificar el <i>concepto</i> de la compra, si se trata de una compra que <i>se repite mensualmente</i> , y el fichero adjunto. Cuando no se cumplan los requisitos de modificación, el sistema mantendrá los datos antiguos y los mostrará de nuevo, informando de los errores.		

Tabla 3.23: RF-22. Modificación de compra

<i>Identificador</i>	RF-23	<i>Título</i>	Eliminación de compra
<i>Módulo</i>	Compras	<i>Prioridad</i>	Baja
<i>Descripción</i>	El sistema, permitirá eliminar definitivamente una compra.		
<i>Observaciones</i>	El sistema pedirá confirmación para eliminar una compra. No se podrá eliminar una compra que ha iniciado su amortización. Una compra que se elimine de esta manera, eliminará también la transacción de pago asociada.		

Tabla 3.24: RF-23. Eliminación de compra

<i>Identificador</i>	RF-24	<i>Título</i>	Gráfica de compras
<i>Módulo</i>	Compras	<i>Prioridad</i>	Media
<i>Descripción</i>	El sistema, mostrará una gráfica del sumario de compras de la empresa.		
<i>Observaciones</i>	El gráfico mostrará el porcentaje de compras agrupadas por: el proveedor de la compra, el destino de la compra (a qué/quién está asociada) o su porcentaje amortizado vs. pendiente.		

Tabla 3.25: RF-24. Gráfica de compras

<i>Identificador</i>	RF-25	<i>Título</i>	Módulo de ventas
<i>Módulo</i>	Ventas	<i>Prioridad</i>	Alta
<i>Descripción</i>	Toda la lógica de negocio relativa a la facturación y los clientes debe estar contenida dentro de una única vista.		
<i>Observaciones</i>	El módulo debe ser seleccionable desde una pestaña (tab).		

Tabla 3.26: RF-25. Módulo de ventas

<i>Identificador</i>	RF-26	<i>Título</i>	Listado de clientes
<i>Módulo</i>	Ventas	<i>Prioridad</i>	Alta
<i>Descripción</i>	El módulo de ventas mostrará un listado, ordenado y paginado de todos los clientes de la empresa.		
<i>Observaciones</i>	El listado se podrá ordenar por cualquier columna.		

Tabla 3.27: RF-26. Listado de clientes

<i>Identificador</i>	RF-27	<i>Título</i>	Alta de cliente
<i>Módulo</i>	Ventas	<i>Prioridad</i>	Alta
<i>Descripción</i>	El módulo de ventas tendrá un botón, que abrirá el formulario de creación de un cliente.		
<i>Observaciones</i>	Al guardar un cliente, el sistema informará de los campos requeridos no completados, y de aquellos que contengan errores.		

Tabla 3.28: RF-27. Alta de cliente

<i>Identificador</i>	RF-28	<i>Título</i>	Modificación de cliente
<i>Módulo</i>	Ventas	<i>Prioridad</i>	Alta
<i>Descripción</i>	El sistema, mostrará el formulario de modificación de cliente, haciendo click en alguno de los campos del listado de clientes.		
<i>Observaciones</i>	Cuando no se cumplan los requisitos de modificación, el sistema mantendrá los datos antiguos y los mostrará de nuevo, informando de los errores.		

Tabla 3.29: RF-28. Modificación de cliente

<i>Identificador</i>	RF-29	<i>Título</i>	Eliminación de cliente
<i>Módulo</i>	Ventas	<i>Prioridad</i>	Baja
<i>Descripción</i>	El sistema, permitirá eliminar definitivamente un cliente.		
<i>Observaciones</i>	El sistema pedirá confirmación para eliminar un cliente. No se podrá eliminar un cliente en uso.		

Tabla 3.30: RF-29. Eliminación de cliente

<i>Identificador</i>	RF-30	<i>Título</i>	Listado de facturas emitidas
<i>Módulo</i>	Ventas	<i>Prioridad</i>	Alta
<i>Descripción</i>	El módulo de ventas mostrará un listado, ordenado y paginado de todas las facturas emitidas por la empresa.		
<i>Observaciones</i>	El listado solo mostrará aquellas facturas con fecha de cobro o, en su defecto, aquellas cuya fecha de vencimiento sea anterior a la fecha actual. El listado se podrá ordenar por cualquier columna. Junto a cada factura se mostrará la <i>fecha de emisión</i> y el <i>importe</i> facturado.		

Tabla 3.31: RF-30. Listado de facturas emitidas

<i>Identificador</i>	RF-31	<i>Título</i>	Listado de facturas pendientes de cobro
<i>Módulo</i>	Ventas	<i>Prioridad</i>	Alta
<i>Descripción</i>	El módulo de ventas mostrará un listado, ordenado y paginado de todas las facturas, pendientes de cobro, emitidas por la empresa.		
<i>Observaciones</i>	El listado solo mostrará aquellas facturas sin fecha de cobro, cuya fecha de vencimiento sea superior a la fecha actual. El listado se podrá ordenar por cualquier columna. Junto a cada factura se mostrará la <i>fecha de vencimiento</i> , el <i>retraso</i> , el <i>cliente</i> y el <i>importe</i> facturado.		

Tabla 3.32: RF-31. Listado de facturas pendientes de cobro

<i>Identificador</i>	RF-32	<i>Título</i>	Alta de Factura
<i>Módulo</i>	Ventas	<i>Prioridad</i>	Alta
<i>Descripción</i>	El módulo de ventas tendrá un botón, que abrirá el formulario de creación de una factura.		
<i>Observaciones</i>	<p>Una factura estará asociada a un cliente. El formulario de creación de factura mostrará un botón junto al selector de cliente, que desplegará el formulario de creación de cliente y permitirá crearlo desde ahí. El cliente creado de esta manera será seleccionado automáticamente como el cliente de la compra. El proveedor de una factura siempre será la empresa usuario. Una factura podrá estar asociada opcionalmente a un proyecto. Si se asocia una factura a un proyecto, su importe se añadirá al total cobrado del proyecto. La fecha de emisión, de vencimiento y de pago pueden diferir, pero la fecha de pago no puede ser anterior a la de vencimiento, ni esta a la de emisión. La cuota de IVA de la factura no se tendrá en cuenta en el flujo de caja. Cada factura permitirá adjuntar un fichero. Al guardar una factura, el sistema informará de los campos requeridos no completados, y de aquellos que contengan errores. La creación de una factura, implicará la creación de una transacción de cobro, coincidente con la fecha de cobro (en su defecto, la fecha de vencimiento) y el importe de la misma.</p>		

Tabla 3.33: RF-32. Alta de factura

<i>Identificador</i>	RF-33	<i>Título</i>	Modificación de factura
<i>Módulo</i>	Ventas	<i>Prioridad</i>	Alta
<i>Descripción</i>	El sistema, mostrará el formulario de modificación de factura, haciendo click en alguno de los campos del listado de facturas.		
<i>Observaciones</i>	<p>Si se modifica el proyecto de una factura, se actualizarán los cálculos de importes cobrados de los proyectos afectados. Si se modifica la fecha de cobro de una factura, se deberá actualizar su transacción de cobro asociada para que su fecha coincida con esta última. Cuando no se cumplan los requisitos de modificación, el sistema mantendrá los datos antiguos y los mostrará de nuevo, informando de los errores.</p>		

Tabla 3.34: RF-33. Modificación de factura

<i>Identificador</i>	RF-34	<i>Título</i>	Eliminación de factura
<i>Módulo</i>	Ventas	<i>Prioridad</i>	Baja
<i>Descripción</i>	El sistema, permitirá eliminar definitivamente una factura.		
<i>Observaciones</i>	El sistema pedirá confirmación para eliminar una factura. Una factura que se elimine de esta manera, eliminará también la transacción de cobro asociada.		

Tabla 3.35: RF-34. Eliminación de factura

<i>Identificador</i>	RF-35	<i>Título</i>	Gráfica de facturación trimestral
<i>Módulo</i>	Ventas	<i>Prioridad</i>	Media
<i>Descripción</i>	El sistema, mostrará una gráfica de la facturación trimestral de la empresa agrupada por ejercicios económicos.		
<i>Observaciones</i>	El gráfico mostrará el total facturado en cada trimestre para cada año. Existirá un selector que permitirá revisar la facturación de otros años.		

Tabla 3.36: RF-35. Gráfica de facturación trimestral

<i>Identificador</i>	RF-36	<i>Título</i>	Módulo de proyectos
<i>Módulo</i>	Proyectos	<i>Prioridad</i>	Alta
<i>Descripción</i>	Toda la lógica de negocio relativa a los proyectos debe estar contenida dentro de una única vista.		
<i>Observaciones</i>	El módulo debe ser seleccionable desde una pestaña (tab).		

Tabla 3.37: RF-36. Módulo de proyectos

<i>Identificador</i>	RF-37	<i>Título</i>	Listado de proyectos
<i>Módulo</i>	Proyectos	<i>Prioridad</i>	Alta
<i>Descripción</i>	El módulo de proyectos mostrará un listado, ordenado y paginado de todos los proyectos y prestaciones de servicio llevadas a cabo por la empresa.		
<i>Observaciones</i>	El listado se podrá ordenar por cualquier columna. Junto a cada proyecto se mostrará el <i>cliente</i> , el <i>presupuesto</i> , el <i>coste</i> total, el importe <i>cobrado</i> , el <i>beneficio</i> neto y el <i>coste</i> de los worklogs del mes en curso.		

Tabla 3.38: RF-37. Listado de proyectos

<i>Identificador</i>	RF-38	<i>Título</i>	Alta de Proyecto
<i>Módulo</i>	Proyectos	<i>Prioridad</i>	Alta
<i>Descripción</i>	El módulo de proyectos tendrá un botón, que abrirá el formulario de creación de un proyecto.		
<i>Observaciones</i>	Un proyecto podrá estar asociado a un cliente. El formulario de creación de proyecto mostrará un botón junto al selector de cliente, que desplegará el formulario de creación de cliente y permitirá crearlo desde ahí. El cliente creado de esta manera será seleccionado automáticamente como el cliente del proyecto. Al guardar un proyecto, el sistema informará de los campos requeridos no completados, y de aquellos que contengan errores.		

Tabla 3.39: RF-38. Alta de proyecto

<i>Identificador</i>	RF-39	<i>Título</i>	Modificación de proyecto
<i>Módulo</i>	Proyectos	<i>Prioridad</i>	Alta
<i>Descripción</i>	El sistema, mostrará el formulario de modificación de proyecto, haciendo click en alguno de los campos del listado de proyectos.		
<i>Observaciones</i>	Si se modifican los datos de un proyecto, se actualizarán los cálculos de importes del proyecto. Los campos, <i>coste</i> y <i>beneficio</i> de un proyecto no se podrán modificar. Cuando no se cumplan los requisitos de modificación, el sistema mantendrá los datos antiguos y los mostrará de nuevo, informando de los errores.		

Tabla 3.40: RF-39. Modificación de proyecto

<i>Identificador</i>	RF-40	<i>Título</i>	Eliminación de proyecto
<i>Módulo</i>	Proyectos	<i>Prioridad</i>	Baja
<i>Descripción</i>	El sistema, permitirá eliminar definitivamente un proyecto.		
<i>Observaciones</i>	El sistema pedirá confirmación para eliminar un proyecto. No será posible eliminar un proyecto que tenga worklogs o haya sido asignado como destino de una compra.		

Tabla 3.41: RF-40. Eliminación de proyecto

<i>Identificador</i>	RF-41	<i>Título</i>	Cálculo de costes de un proyecto
<i>Módulo</i>	Proyectos	<i>Prioridad</i>	Alta
<i>Descripción</i>	Los campos que acompañan a cada proyecto en el listado de proyectos de la empresa, deben ser calculados, cada vez que se modifica un proyecto, worklog o se realiza una compra asociada a uno de ellos, o se emite una factura asociada a uno de ellos.		
<i>Observaciones</i>	<p>Cada campo se calculará de la siguiente manera:</p> <ul style="list-style-type: none"> ▪ Presupuesto: introducido por el usuario manualmente. ▪ Coste: calculado de la suma de amortizaciones mensuales de las compras asociadas al proyecto, más el sumatorio de todos los worklogs del proyecto en base a: multiplicar la duración en horas del worklog por el coste/hora del empleado que lo realizó, de la paga del mes en el que se llevó a cabo el worklog. ▪ Cobrado: calculado de la suma de importes de las facturas asociadas al proyecto, con fecha de cobro anterior o igual a la fecha actual. ▪ Beneficio: resultante de restar el coste total, del total cobrado del proyecto. ▪ Coste mes: similar al coste total, pero solo tiene en cuenta los worklogs del mes en curso. 		

Tabla 3.42: RF-41. Cálculo de costes de un proyecto

<i>Identificador</i>	RF-42	<i>Título</i>	Gráfica de sumario de proyectos
<i>Módulo</i>	Proyectos	<i>Prioridad</i>	Media
<i>Descripción</i>	El sistema, mostrará una gráfica de sumarización de los proyectos de la empresa.		
<i>Observaciones</i>	El gráfico mostrará el porcentaje de proyectos por cliente / tiempo empleado.		

Tabla 3.43: RF-42. Gráfica de sumario de proyectos

<i>Identificador</i>	RF-43	<i>Título</i>	Gráfica de situación de proyectos
<i>Módulo</i>	Proyectos	<i>Prioridad</i>	Media
<i>Descripción</i>	El sistema, mostrará una gráfica de situación de los proyectos de la empresa.		
<i>Observaciones</i>	El gráfico mostrará el porcentaje de beneficio vs. coste y el porcentaje presu- puestado vs. cobrado de los proyectos de la empresa.		

Tabla 3.44: RF-43. Gráfica de situación de proyectos

<i>Identificador</i>	RF-44	<i>Título</i>	Listado de últimos worklogs
<i>Módulo</i>	Proyectos	<i>Prioridad</i>	Alta
<i>Descripción</i>	El formulario de modificación de un proyecto, mostrará un listado, ordenado y paginado de los últimos worklogs (o reportes de trabajo) creados por los empleados.		
<i>Observaciones</i>	El listado se podrá ordenar por cualquier columna. El listado estará ordenado por fecha de manera descendente.		

Tabla 3.45: RF-44. Listado de últimos worklogs

<i>Identificador</i>	RF-45	<i>Título</i>	Alta de worklog
<i>Módulo</i>	Proyectos	<i>Prioridad</i>	Alta
<i>Descripción</i>	El listado de últimos worklogs tendrá un botón, que abrirá el formulario de creación de un worklog.		
<i>Observaciones</i>	Un worklog será identificado de manera unívoca por su clave <i>Issue Key</i> . Estará asociado a un empleado obligatoriamente. El campo tiempo empleado seguirá un pseudo-lenguaje natural similar al de Jira (w d h m). Con la creación de cada nuevo worklog, se deberán actualizar los costes totales del proyecto al que pertenece. Al guardar un worklog, el sistema informará de los campos requeridos no completados, y de aquellos que contengan errores.		

Tabla 3.46: RF-45. Alta de worklog

<i>Identificador</i>	RF-46	<i>Título</i>	Modificación de worklog
<i>Módulo</i>	Proyectos	<i>Prioridad</i>	Alta
<i>Descripción</i>	El sistema, mostrará el formulario de modificación de un worklog, haciendo click en alguno de los campos del listado.		
<i>Observaciones</i>	Con la modificación de cada worklog, se deberán actualizar los costes totales del proyecto al que pertenece. Cuando no se cumplan los requisitos de modificación, el sistema mantendrá los datos antiguos y los mostrará de nuevo, informando de los errores.		

Tabla 3.47: RF-46. Modificación de worklog

<i>Identificador</i>	RF-47	<i>Título</i>	Eliminación de worklog
<i>Módulo</i>	Proyectos	<i>Prioridad</i>	Media
<i>Descripción</i>	El sistema, permitirá eliminar definitivamente un worklog.		
<i>Observaciones</i>	El sistema pedirá confirmación para eliminar un worklog. Tras eliminar un worklog se recalcularán los valores del proyecto.		

Tabla 3.48: RF-47. Eliminación de worklog

<i>Identificador</i>	RF-48	<i>Título</i>	Módulo de balance
<i>Módulo</i>	Balance	<i>Prioridad</i>	Alta
<i>Descripción</i>	Toda la lógica de negocio relativa a la tesorería de la empresa, debe estar contenida dentro de una única vista.		
<i>Observaciones</i>	El módulo debe ser seleccionable desde una pestaña (tab).		

Tabla 3.49: RF-48. Módulo de balance

<i>Identificador</i>	RF-49	<i>Título</i>	Previsión de pagos
<i>Módulo</i>	Balance	<i>Prioridad</i>	Alta
<i>Descripción</i>	El módulo de balance mostrará un listado, ordenado y paginado de todas las transacciones de pago que tiene que hacer frente la empresa.		
<i>Observaciones</i>	El listado se podrá ordenar por cualquier columna. De cada transacción se mostrará la <i>fecha</i> prevista del pago, el <i>concepto</i> y el importe.		

Tabla 3.50: RF-49. Previsión de pagos

<i>Identificador</i>	RF-50	<i>Título</i>	Alta de pago
<i>Módulo</i>	Balance	<i>Prioridad</i>	Alta
<i>Descripción</i>	El módulo de balance tendrá un botón, que abrirá el formulario de creación de una transacción de pago.		
<i>Observaciones</i>	Aunque el sistema permitirá crear manualmente transacciones de pago, la mayoría de éstas, serán generadas por el sistema de manera automática, al introducir una compra en el sistema, o bien al generar las operaciones de pago de nóminas o cuotas de la seguridad social durante el proceso de cambio de mes. Al guardar un pago, el sistema informará de los campos requeridos no completados, y de aquellos que contengan errores.		

Tabla 3.51: RF-50. Alta de pago

<i>Identificador</i>	RF-51	<i>Título</i>	Modificación de pago
<i>Módulo</i>	Balance	<i>Prioridad</i>	Alta
<i>Descripción</i>	El sistema, mostrará el formulario de modificación de transacción de pago, haciendo click en alguno de los campos del listado de previsión de pagos, o al pulsar sobre la etiqueta que identifique el pago en el gráfico de previsión de caja.		
<i>Observaciones</i>	Cuando no se cumplan los requisitos de modificación, el sistema mantendrá los datos antiguos y los mostrará de nuevo, informando de los errores.		

Tabla 3.52: RF-51. Modificación de pago

<i>Identificador</i>	RF-52	<i>Título</i>	Eliminación de pago
<i>Módulo</i>	Balance	<i>Prioridad</i>	Baja
<i>Descripción</i>	El sistema, permitirá eliminar definitivamente un pago.		
<i>Observaciones</i>	El sistema pedirá confirmación para eliminar un pago.		

Tabla 3.53: RF-52. Eliminación de pago

<i>Identificador</i>	RF-53	<i>Título</i>	Previsión de cobros
<i>Módulo</i>	Balance	<i>Prioridad</i>	Alta
<i>Descripción</i>	El módulo de balance mostrará un listado, ordenado y paginado de todas las transacciones de cobro que tiene que previsto hacer la empresa.		
<i>Observaciones</i>	El listado se podrá ordenar por cualquier columna. De cada transacción se mostrará la <i>fecha</i> prevista del cobro, el número de <i>factura</i> y el importe.		

Tabla 3.54: RF-53. Previsión de cobros

<i>Identificador</i>	RF-54	<i>Título</i>	Alta de cobro
<i>Módulo</i>	Balance	<i>Prioridad</i>	Alta
<i>Descripción</i>	El módulo de balance tendrá un botón, que abrirá el formulario de creación de una transacción de cobro.		
<i>Observaciones</i>	Aunque el sistema permitirá crear manualmente transacciones de cobro, la mayoría de éstas, serán generadas por el sistema de manera automática, al introducir una factura en el sistema. Al guardar un cobro, el sistema informará de los campos requeridos no completados, y de aquellos que contengan errores.		

Tabla 3.55: RF-54. Alta de cobro

<i>Identificador</i>	RF-55	<i>Título</i>	Modificación de cobro
<i>Módulo</i>	Balance	<i>Prioridad</i>	Alta
<i>Descripción</i>	El sistema, mostrará el formulario de modificación de transacción de cobro, haciendo click en alguno de los campos del listado de previsión de cobros, o al pulsar sobre la etiqueta que identifique el cobro en el gráfico de previsión de caja.		
<i>Observaciones</i>	Cuando no se cumplan los requisitos de modificación, el sistema mantendrá los datos antiguos y los mostrará de nuevo, informando de los errores.		

Tabla 3.56: RF-55. Modificación de cobro

<i>Identificador</i>	RF-56	<i>Título</i>	Eliminación de cobro
<i>Módulo</i>	Balance	<i>Prioridad</i>	Baja
<i>Descripción</i>	El sistema, permitirá eliminar definitivamente un cobro.		
<i>Observaciones</i>	El sistema pedirá confirmación para eliminar un cobro.		

Tabla 3.57: RF-56. Eliminación de cobro

<i>Identificador</i>	RF-57	<i>Título</i>	Previsión de impuestos
<i>Módulo</i>	Balance	<i>Prioridad</i>	Alta
<i>Descripción</i>	El módulo de balance mostrará un listado, ordenado y paginado de todos los pagos de impuestos a los que la empresa tiene que hacer frente trimestralmente.		
<i>Observaciones</i>	El listado se podrá ordenar por cualquier columna. De cada impuesto se mostrará el <i>ejercicio</i> y <i>trimestre</i> al que pertenece, el <i>concepto</i> , <i>importe</i> y la <i>fecha</i> de pago prevista.		

Tabla 3.58: RF-57. Previsión de impuestos

<i>Identificador</i>	RF-58	<i>Título</i>	Alta de impuesto
<i>Módulo</i>	Balance	<i>Prioridad</i>	Alta
<i>Descripción</i>	El módulo de balance tendrá un botón, que abrirá el formulario de creación del pago de un impuesto.		
<i>Observaciones</i>	La fecha de pago del impuesto vendrá dada por el trimestre al que pertenece, esta fecha se definirá en la configuración de la aplicación. Al guardar un cobro, el sistema informará de los campos requeridos no completados, y de aquellos que contengan errores.		

Tabla 3.59: RF-58. Alta de impuesto

<i>Identificador</i>	RF-59	<i>Título</i>	Modificación de impuesto
<i>Módulo</i>	Balance	<i>Prioridad</i>	Alta
<i>Descripción</i>	El sistema, mostrará el formulario de modificación del pago de un impuesto, haciendo click en alguno de los campos del listado de previsión de impuestos, o al pulsar sobre la etiqueta que identifique el impuesto en el gráfico de previsión de caja.		
<i>Observaciones</i>	Cuando no se cumplan los requisitos de modificación, el sistema mantendrá los datos antiguos y los mostrará de nuevo, informando de los errores.		

Tabla 3.60: RF-59. Modificación de impuesto

<i>Identificador</i>	RF-60	<i>Título</i>	Eliminación de impuesto
<i>Módulo</i>	Balance	<i>Prioridad</i>	Baja
<i>Descripción</i>	El sistema, permitirá eliminar definitivamente un impuesto.		
<i>Observaciones</i>	El sistema pedirá confirmación para eliminar un impuesto.		

Tabla 3.61: RF-60. Eliminación de impuesto

<i>Identificador</i>	RF-61	<i>Título</i>	Gráfica de balance de caja
<i>Módulo</i>	Balance	<i>Prioridad</i>	Media
<i>Descripción</i>	El sistema, mostrará una gráfica de variación del flujo de caja la empresa.		
<i>Observaciones</i>	El gráfico mostrará un flujo temporal del saldo de caja de la empresa entre dos fechas dadas. Por defecto el gráfico mostrará todos los movimientos de caja para el ejercicio en curso, pero permitirá filtrar por cualquier combinación de fechas. Cada transacción del gráfico permitirá visualizar tanto su variación de saldo como el detalle de la naturaleza de la transacción (pago, cobro, factura, compra o impuesto).		

Tabla 3.62: RF-61. Gráfica de balance de caja

<i>Identificador</i>	RF-62	<i>Título</i>	Proceso de cambio de mes
<i>Módulo</i>		<i>Prioridad</i>	Alta
<i>Descripción</i>	El sistema, al iniciar un nuevo mes, deberá realizar una serie de tareas para cerrar la contabilidad del mes en curso, y preparar el próximo mes.		
<i>Observaciones</i>	Este proceso incluirá las tareas de <i>copiar pagas del mes anterior</i> , <i>amortizar todas las compras</i> un no amortizadas completamente, generar las transacciones de <i>pago de nóminas y cuotas de la seguridad social</i> , y por último, copiar las <i>compras recurrentes</i> .		

Tabla 3.63: RF-62. Proceso de cambio de mes

<i>Identificador</i>	RF-63	<i>Título</i>	Amortizar las compras
<i>Módulo</i>		<i>Prioridad</i>	Alta
<i>Descripción</i>	El sistema, al iniciar un nuevo mes, calculará el porcentaje amortizado de cada compra que no esté completamente amortizada.		
<i>Observaciones</i>	<p>El porcentaje amortizado, se calculará dividiendo el importe total de la compra (sin la cuota de IVA) entre el periodo de amortización, dando como resultado la cuota mensual. Esta cuota mensual, será repartida según el destino de la compra (<i>asociado a</i>) de la siguiente manera:</p> <ul style="list-style-type: none"> ▪ Empleado: la paga del mes en curso, del empleado al que está asociada la compra, asume todo el importe de la cuota mensual amortizada. ▪ Proyecto: el coste del proyecto al que está asociada la compra, asume todo el importe de la cuota mensual amortizada. ▪ Empresa: si la compra está asociada a la empresa usuario, se divide el importe de la cuota mensual amortizada entre las pagas mensuales de todos los empleados de manera equitativa según su salario mensual. 		

Tabla 3.64: RF-63. Amortizar las compras

<i>Identificador</i>	RF-64	<i>Título</i>	Copiar pagas mes anterior
<i>Módulo</i>		<i>Prioridad</i>	Alta
<i>Descripción</i>	El sistema, al iniciar un nuevo mes, deberá copiar las pagas del mes anterior de los empleados <i>activos</i> .		
<i>Observaciones</i>	La propiedades de esta nueva paga serán iguales a las del mes anterior, con la nueva fecha y sus costes asumidos a cero.		

Tabla 3.65: RF-64. Copiar pagas mes anterior

<i>Identificador</i>	RF-65	<i>Título</i>	Generar pagos de nóminas y SS
<i>Módulo</i>		<i>Prioridad</i>	Alta
<i>Descripción</i>	El sistema, al iniciar un nuevo mes, calculará el importe total de las nóminas y cuotas de la SS de los empleados activos y generará una transacción de pago, para cada concepto, para el día 28 del nuevo mes.		
<i>Observaciones</i>	Estas transacciones de pago, podrán ser modificadas posteriormente.		

Tabla 3.66: RF-65. Generar pagos de nóminas y SS

<i>Identificador</i>	RF-66	<i>Título</i>	Copiar compras recurrentes
<i>Módulo</i>		<i>Prioridad</i>	Alta
<i>Descripción</i>	El sistema, al iniciar un nuevo mes, copiará aquellas compras marcadas como <i>compras que se repiten</i> , modificando el mes de la fecha de compra convenientemente.		
<i>Observaciones</i>	Esta nueva operación de compra, generará una transacción de pago, con las fechas adecuadas como si la hubiera creado el usuario manualmente.		

Tabla 3.67: RF-66. Copiar compras recurrentes

3.1.3. Requisitos No Funcionales

Un requisito no funcional, o atributo de calidad, es aquél que puede usarse para juzgar la operación de un sistema en lugar de sus comportamientos específicos.

<i>Identificador</i>	RNF-1	<i>Título</i>	Software multiplataforma
<i>Módulo</i>		<i>Prioridad</i>	Alta
<i>Descripción</i>	El sistema deberá poder funciona en cualquier sistema operativo (Linux, Windows, OSX).		
<i>Observaciones</i>			

Tabla 3.68: RNF-1. Software multiplataforma

<i>Identificador</i>	RNF-2	<i>Título</i>	Compatibilidad con navegadores
<i>Módulo</i>		<i>Prioridad</i>	Alta
<i>Descripción</i>	El sistema deberá ofrecer una experiencia similar con cualquier navegador moderno.		
<i>Observaciones</i>	Se requiere como mínimo compatibilidad con Firefox, Chrome y Safari.		

Tabla 3.69: RNF-2. Compatibilidad con navegadores

<i>Identificador</i>	RNF-3	<i>Título</i>	Disponibilidad on-line
<i>Módulo</i>		<i>Prioridad</i>	Alta
<i>Descripción</i>	El sistema deberá ser accesible desde cualquier puesto remoto con acceso a internet.		
<i>Observaciones</i>	El sistema deberá ser por tanto una aplicación web.		

Tabla 3.70: RNF-3. Disponibilidad on-line

<i>Identificador</i>	RNF-4	<i>Título</i>	Usabilidad
<i>Módulo</i>		<i>Prioridad</i>	Alta
<i>Descripción</i>	El sistema mostrará mensajes de error descriptivos, identificando en cada caso la naturaleza del error y dando indicaciones de cómo solventarlo.		
<i>Observaciones</i>			

Tabla 3.71: RNF-4. Usabilidad

<i>Identificador</i>	RNF-5	<i>Título</i>	Multi-idioma
<i>Módulo</i>		<i>Prioridad</i>	Alta
<i>Descripción</i>	El sistema se mostrará por defecto en Español, pero deberá estar preparado para su futura traducción a cualquier lenguaje.		
<i>Observaciones</i>	El sistema deberá ser sensible a la cabecera de lenguaje de los navegadores y mostrar el lenguaje según la preferencia del usuario.		

Tabla 3.72: RNF-5. Multi-idioma

<i>Identificador</i>	RNF-6	<i>Título</i>	Tolerancia a fallos
<i>Módulo</i>		<i>Prioridad</i>	Alta
<i>Descripción</i>	El sistema deberá ser tolerante a fallos provocados por las operaciones que realiza el usuario o aquellas que se ejecutan de manera automática por el sistema.		
<i>Observaciones</i>			

Tabla 3.73: RNF-6. Tolerancia a fallos

<i>Identificador</i>	RNF-7	<i>Título</i>	Experiencia de usuario
<i>Módulo</i>		<i>Prioridad</i>	Alta
<i>Descripción</i>	El sistema utilizará un framework de renderización de pantallas asíncrono (AJAX) para garantizar un tiempo de respuesta mínimo en la navegación de pantallas.		
<i>Observaciones</i>			

Tabla 3.74: RNF-7. Experiencia de usuario

<i>Identificador</i>	RNF-8	<i>Título</i>	Manual de usuario
<i>Módulo</i>		<i>Prioridad</i>	Alta
<i>Descripción</i>	Se proporcionará un manual que documente, cada uno de los componentes que forman la solución propuesta y cómo utilizarlos.		
<i>Observaciones</i>			

Tabla 3.75: RNF-8. Manual de usuario

<i>Identificador</i>	RNF-9	<i>Título</i>	Interfaz de usuario
<i>Módulo</i>		<i>Prioridad</i>	Alta
<i>Descripción</i>	La interfaz de usuario del sistema, deberá tener un diseño limpio y poco intrusivo, que muestre u oculte las ventanas necesarias para su interoperabilidad de manera automática.		
<i>Observaciones</i>			

Tabla 3.76: RNF-9. Interfaz de usuario

3.2. Identificación de Usuarios

En este apartado se identifican los usuarios que harán uso de la aplicación según sus necesidades. Aunque inicialmente no se requirió acceso securizado a la aplicación, al estar desplegada dentro de una vpn con acceso único a los usuarios de la empresa, se planteó como mejora futura la creación de usuarios, roles y perfiles para el acceso y manejo de la herramienta.

3.2.1. Recursos Humanos

El personal de recursos humanos de la empresa, precisará acceder a la aplicación, para reflejar cualquier cambio que pudiera darse con los empleados y sus nóminas: desde gestionar las altas y bajas, a cualquier modificación sobre los datos personales o salarios de los trabajadores, así como verificar que los gastos de personal están correctamente imputados en la aplicación para el cierre del mes.

El módulo de *empleados* será su principal área de actividad.

3.2.2. Contable/Administrativo

El personal del dpto. de contabilidad o gestión administrativa, precisará acceder a la aplicación, para gestionar los proveedores de la empresa, las compras y sus amortizaciones, generar la facturación mensual de los proyectos y servicios prestados, así como mantener reflejado el flujo de caja y la tesorería de la empresa, imputando gastos e impuestos en las fechas previstas.

Los módulos de *compras*, *ventas*, *proyectos* y *balance* serán su principal área de actividad.

3.2.3. Técnico Preventa

El personal del dpto. comercial o técnicos preventa, precisara acceder a la aplicación, para dar de alta los proyectos y prestaciones de servicio que hayan conseguido para la empresa. Igualmente tendrán a su disposición los datos de proyectos y prestaciones de servicio en curso o pasadas, para poder negociar con mayor ventaja competitiva los nuevos presupuestos.

Los módulos de *proyectos* y *empleados* serán su principal área de actividad.

3.2.4. Ingeniero de Software

El personal del dpto. de desarrollo, precisará acceder a la aplicación, para registrar los reportes de trabajo (o worklogs) que posteriormente servirán para calcular los costes de un proyecto (y por ende su beneficio).

3.2.5. Administrador

El administrador, será un perfil experto en la herramienta, que tendrá la capacidad de realizar tareas complejas de la aplicación, tales como definir los valores de configuración o realizar tareas correctoras como *eliminar las pagas del mes en curso* o *des-amortizar las compras del mes en curso*. En general, será capaz de dar soporte ante cualquier situación anómala de la aplicación a cualquiera de los otros roles.

3.3. Diagramas de Casos de Uso

Los diagramas de casos de uso, detallan las interacciones (relaciones) entre los usuarios (actores) y los casos de uso (sistema) de la herramienta.

Los casos de uso de la aplicación, se corresponden con los requisitos funcionales descritos en el apartado anterior. Dada su extensión y complejidad, se describirán detalladamente, sólo aquellos que por su relevancia, pueden aportar más claridad al proceso de análisis.

Para detallar los casos de uso se utilizarán la representación gráfica (diagramas) y la representación escrita (tablas), utilizando la plantilla siguiente:

3.3.1. Plantilla

Los diagramas de casos de uso, se representarán mediante 4 elementos gráficos:

- Actores. Se le llama actor a toda entidad externa al sistema que guarda una relación con éste y que le demanda una funcionalidad. Esto incluye a los operadores humanos pero también incluye a todos los sistemas externos, además de entidades abstractas, como el tiempo. Se representará con la siguiente figura:



Figura 3.1: Diagramas de casos de uso: Representación de actor

- Caso de uso. Se llama caso de uso a la acción o funcionalidad que se realiza por interacción del usuario en un determinado escenario. Se representará con la siguiente figura:



Figura 3.2: Diagramas de casos de uso: Representación de caso de uso

- Relaciones. Se llama relación de un caso de uso con un actor o con otro caso de uso al tipo de interacción entre estos elementos.

Asociación. Indica la participación de un actor en el caso de uso. Se representa con la siguiente figura:



Figura 3.3: Diagramas de casos de uso: Representación de relación de asociación

Inclusión. Indica la dependencia de un caso de uso con el comportamiento de otro, por estar incluido en éste. Se representa con la siguiente figura:

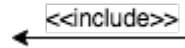


Figura 3.4: Diagramas de casos de uso: Representación de relación de inclusión

Especialización. Indica la dependencia de un caso de uso con el comportamiento de otro, por ser especialización de éste. Se representa con la siguiente figura:

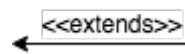


Figura 3.5: Diagramas de casos de uso: Representación de relación de especialización

- Escenario. Delimita el entorno del escenario. Se representa con la siguiente figura:

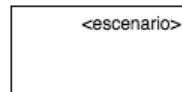


Figura 3.6: Diagramas de casos de uso: Representación de escenario

Los diagramas de casos de uso, se detallarán mediante la tabla:

<i>Identificador</i>	{identificador}	<i>Objetivo</i>	{objetivo}
<i>Escenario</i>	{escenario}	<i>Actores</i>	{actores}
<i>Descripción</i>	{descripción}		
<i>Pre-condiciones</i>	{precondiciones}		
<i>Post-condiciones</i>	{postcondiciones}		

Tabla 3.77: Plantilla de casos de uso

Leyenda:

- Identificador: Identificador único del caso de uso. Compuesto por un prefijo y una clave numérica.
- Objetivo: actividad a realizar por el/los actores.

- Escenario: módulo o funcionalidad de la aplicación en el que se encuadra el caso de uso. Puede ser de sistema y no pertenecer a un módulo particular.
- Actores: el sujeto que realiza la actividad, puede ser alguno de los identificados, o el tiempo.
- Descripción: breve texto explicativo del caso de uso.
- Pre-condiciones: condiciones necesarias para la ocurrencia del caso de uso.
- Post-condiciones: condiciones provocadas por la ocurrencia del caso de uso.

3.3.2. Caso de uso general

En el caso de uso general, observamos todos los casos generales de la aplicación, los actores, y sus relaciones con cada uno de ellos. Ese caso da un aspecto general de la interacción de los usuarios con la aplicación según su rol.

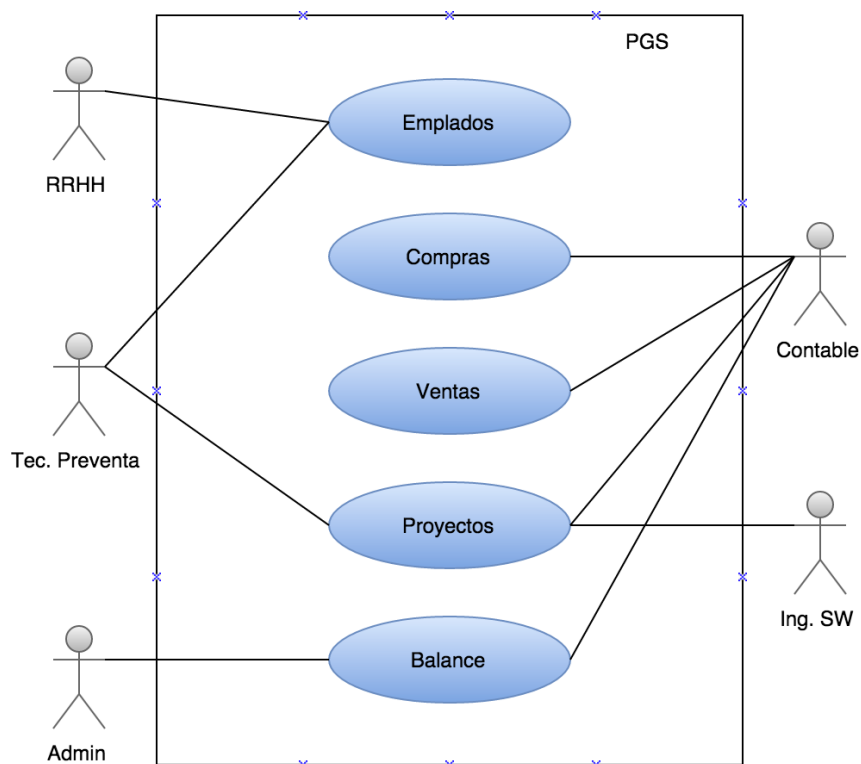


Figura 3.7: Diagramas de casos de uso: Caso de uso general

Identificador	UC-1	Objetivo	Uso de la aplicación
Escenario	General	Actores	RRHH, Cont., Tec. Preventa, Ing. de SW, Admin.
Descripción	Acceder a cada módulo de funcionalidad de la aplicación		
Pre-condiciones	La aplicación debe estar desplegada en un entorno con acceso a internet y securizado por una vpn.		
Post-condiciones			

Tabla 3.78: UC-1. Caso de uso general

3.3.3. Casos de uso del módulo de empleados

Si se explota el caso de uso *empleados* del caso de uso general, observamos los casos de uso del módulo de empleados detallados a continuación.

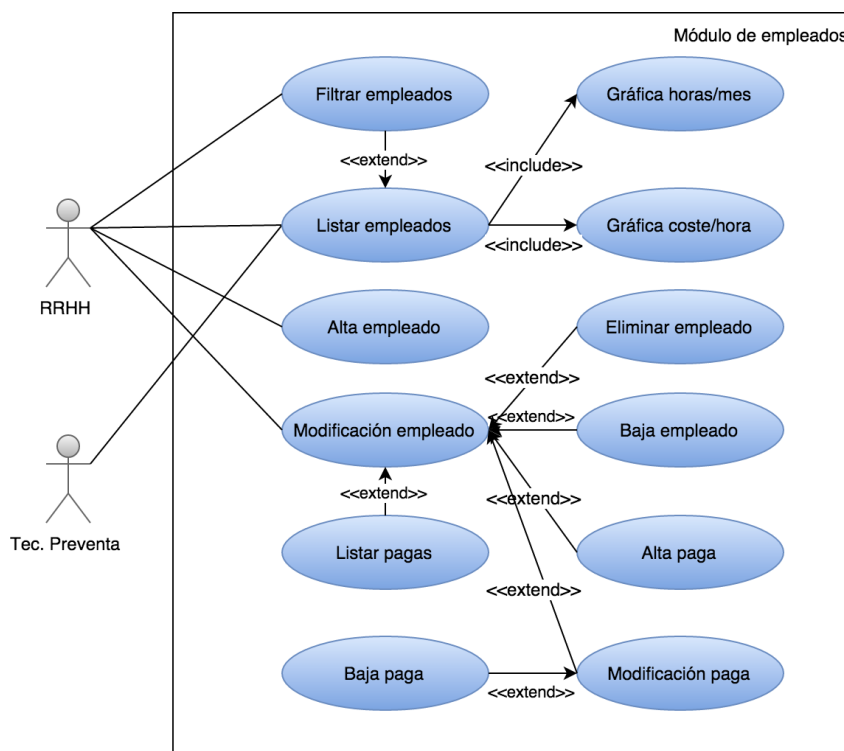


Figura 3.8: Diagramas de casos de uso: Módulo de empleados

Para el escenario del módulo de empleados se han decidido detallar los casos de uso de “Listar empleados”, “Baja empleado”, “Eliminar empleado” y “Gráfica horas/mes”.

El caso de uso “Filtrar empleados” es un claro caso de relación de especialización del caso de uso “Listar empleados”, tal como muestra la figura 3.8.

<i>Identificador</i>	UC-2	<i>Objetivo</i>	Filtrar empleados
<i>Escenario</i>	Módulo de empleados	<i>Actores</i>	RRHH, Tec. Preventa
<i>Descripción</i>	Filtrar los empleados de la aplicación según su estado (activo, inactivo, todos). Los pasos a seguir serán: 1. Acceder al módulo de empleados haciendo click en la pestaña <i>empleados</i> . 2. Hacer click en alguno de los enlaces junto al título del <i>listado de empleados</i> .		
<i>Pre-condiciones</i>	Debe existir al menos un empleado creado en el sistema para poder filtrar el listado.		
<i>Post-condiciones</i>	El listado de empleados solo debe mostrar aquellos empleados cuyo estado coincida con el filtrado.		

Tabla 3.79: UC-2. Filtrar empleados

El caso de uso “Baja empleado” es otro claro ejemplo de relación de especialización, en esta ocasión del caso de uso “Modificación empleado”.

<i>Identificador</i>	UC-3	<i>Objetivo</i>	Baja empleado
<i>Escenario</i>	Módulo de empleados	<i>Actores</i>	RRHH
<i>Descripción</i>	Dar de baja un empleado activo. Los pasos a seguir serán: 1. Acceder al módulo de empleados haciendo click en la pestaña <i>empleados</i> . 2. Hacer click en el <i>nombre</i> de algún empleado del <i>listado de empleados</i> para mostrar su formulario de modificación. 3. Hacer click en el botón <i>dar de baja</i> . 4. Confirmar la acción.		
<i>Pre-condiciones</i>	Debe existir al menos un empleado creado en el sistema para poder darlo de baja. El empleado debe estar en estado activo para poderlo dar de baja.		
<i>Post-condiciones</i>	El empleado quedará en estado <i>baja</i> .		

Tabla 3.80: UC-3. Baja empleado

El caso de uso “Eliminación empleado” es otro claro ejemplo de relación de especialización del caso “Modificación empleado”.

<i>Identificador</i>	UC-4	<i>Objetivo</i>	Eliminar empleado
<i>Escenario</i>	Módulo de empleados	<i>Actores</i>	RRHH
<i>Descripción</i>	Eliminar definitivamente un empleado. Los pasos a seguir serán: 1. Acceder al módulo de empleados haciendo click en la pestaña <i>empleados</i> . 2. Filtrar el listado haciendo click en el enlace “inactivos” junto al título del <i>listado de empleados</i> . 2. Hacer click en el <i>nombre</i> de algún empleado del <i>listado de empleados</i> para mostrar su formulario de modificación. 3. Hacer click en el botón <i>eliminar</i> . 4. Confirmar la acción.		
<i>Pre-condiciones</i>	Debe existir al menos un empleado creado en el sistema. El empleado debe estar en estado baja para poderlo eliminar. El empleado no estará referenciado como destino de una compra. El empleado no tendrá worklogs existentes en ningún proyecto.		
<i>Post-condiciones</i>	El empleado será borrado completamente del sistema.		

Tabla 3.81: UC-4. Eliminar empleado

<i>Identificador</i>	UC-5	<i>Objetivo</i>	Gráfica horas/mes
<i>Escenario</i>	Módulo de empleados	<i>Actores</i>	RRHH, Tec. Prev.
<i>Descripción</i>	Mostrar un gráfico con el número de horas mensuales trabajadas por cada empleado. Los pasos a seguir serán: 1. Acceder al módulo de empleados haciendo click en la pestaña <i>empleados</i> . 2. La gráfica aparecerá sobre el listado de empleados.		
<i>Pre-condiciones</i>	Debe existir al menos un empleado creado en el sistema para poder mostrar su valor en la gráfica. El empleado anterior debe tener al menos un worklog imputado en el mes en curso.		
<i>Post-condiciones</i>			

Tabla 3.82: UC-5. Gráfica horas/mes

El caso de uso “Gráfica horas/mes”, detallado en la tabla 3.82, es un claro ejemplo, de caso de uso de relación de inclusión, del caso de uso “Listar empleados”.

3.3.4. Casos de uso del módulo de compras

Si se explota el caso de uso *compras* del caso de uso general, observamos los casos de uso del módulo de compras detallados a continuación.

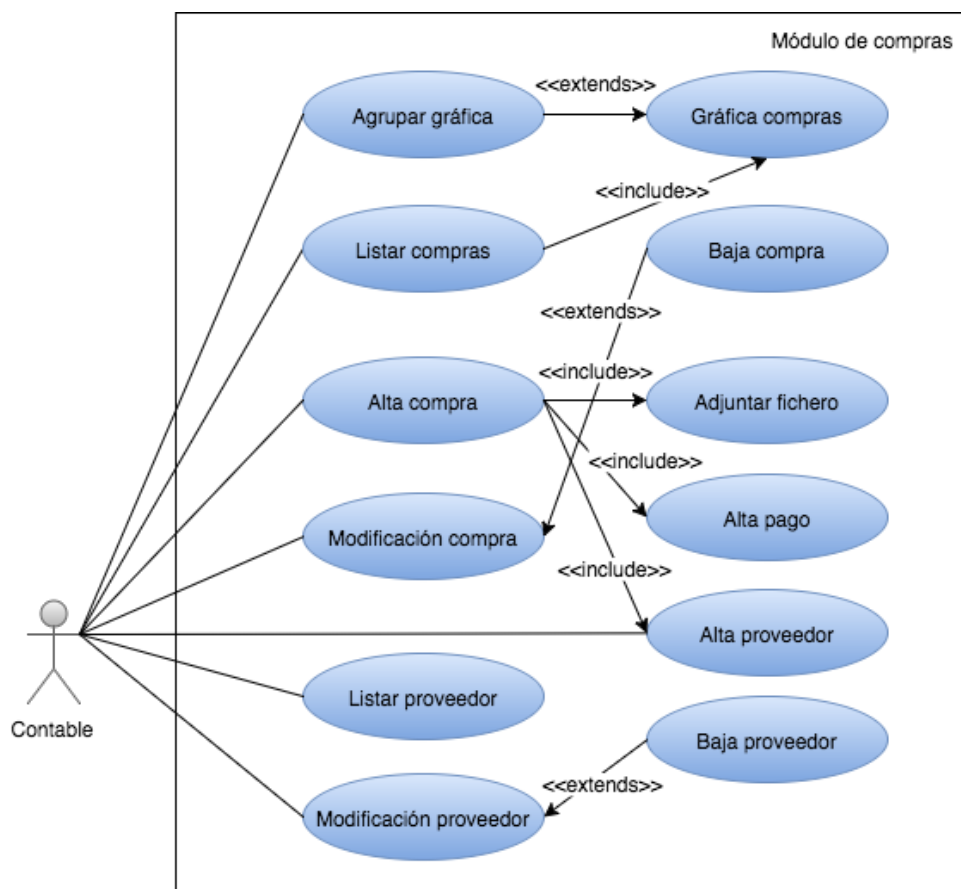


Figura 3.9: Diagramas de casos de uso: Módulo de compras

Para el escenario del módulo de compras se han decidido detallar los casos de uso de “Agrupar gráfica”, “Modificación compra” y “Alta pago”.

El caso de uso “Agrupar gráfica” es un claro caso de relación de especialización del caso de uso “Gráfica compras”, tal como muestra la figura 3.9.

<i>Identificador</i>	UC-6	<i>Objetivo</i>	Agrupar gráfica
<i>Escenario</i>	Módulo de compras	<i>Actores</i>	Contable
<i>Descripción</i>	<p>Modificar la gráfica de compras para que muestre:</p> <ol style="list-style-type: none"> El porcentaje de compras agrupadas por proveedor. El porcentaje de compras agrupadas por el destino de la compra. El porcentaje de amortización pendiente vs. amortizado. <p>Los pasos a seguir serán:</p> <ol style="list-style-type: none"> Acceder al módulo de compras haciendo click en la pestaña <i>compras</i>. Hacer click en el combo selector “Agrupar por” junto a la gráfica “sumario de compras”. 		
<i>Pre-condiciones</i>	Debe existir al menos una compra creada en el sistema para poder agrupar la gráfica.		
<i>Post-condiciones</i>	La gráfica de compras debe mostrar la agrupación correcta de compras.		

Tabla 3.83: UC-6. Agrupar gráfica

El caso de uso “Modificación compra” es un caso de uso, del cual se extiende el caso “Baja compra”, tal como muestra la figura 3.9.

<i>Identificador</i>	UC-7	<i>Objetivo</i>	Modificación compra
<i>Escenario</i>	Módulo de compras	<i>Actores</i>	Contable
<i>Descripción</i>	<p>Modificar una compra existente. Los pasos a seguir serán:</p> <ol style="list-style-type: none"> Acceder al módulo de compras haciendo click en la pestaña <i>compras</i>. Hacer click en el <i>concepto</i> de una compra, del <i>listado de compras</i>, para desplegar el formulario de modificación. Modificar los campos deseados / disponibles. Hacer click en guardar. 		
<i>Pre-condiciones</i>	<p>Debe existir al menos una compra para poder guardar la compra.</p> <p>La compra no debe haber iniciado su amortización, de lo contrario, solo el <i>fichero adjunto</i> y el selector de compra <i>que se repite</i> se podrán editar. Los datos de la compra deben ser válidos.</p>		
<i>Post-condiciones</i>	La compra presentará los nuevos datos guardados.		

Tabla 3.84: UC-7. Modificación compra

El caso de uso “Alta pago” es un claro caso de relación de inclusión del caso de uso “Alta compra”, tal como muestra la figura 3.9.

<i>Identificador</i>	UC-8	<i>Objetivo</i>	Alta pago
<i>Escenario</i>	Módulo de compras	<i>Actores</i>	Contable
<i>Descripción</i>	<p>Crear una transacción de pago asociada a la compra a guardar, de manera automática, haciendo coincidir:</p> <ol style="list-style-type: none"> 1. La fecha del pago, con la fecha de compra o pago de la compra. 2. El importe del pago, con el importe de la compra. 3. El concepto del pago, con el concepto de la compra. <p>Los pasos a seguir serán:</p> <ol style="list-style-type: none"> 1. Acceder al módulo de compras haciendo click en la pestaña <i>compras</i>. 2. Hacer click en el botón <i>añadir</i> para desplegar el formulario de creación. 3. Hacer click en el botón <i>guardar</i>. 4. Se creará el pago de forma automática tras guardar la compra. 		
<i>Pre-condiciones</i>	<p>Debe existir al menos un proveedor para poder guardar la compra.</p> <p>Los datos de la compra deben ser válidos.</p>		
<i>Post-condiciones</i>	<p>El módulo de balance presentará un nuevo pago que coincide con los datos de la compra.</p>		

Tabla 3.85: UC-8. Alta pago

3.3.5. Casos de uso del módulo de ventas

Si se explota el caso de uso *ventas* del caso de uso general, observamos los casos de uso del módulo de ventas detallados a continuación.

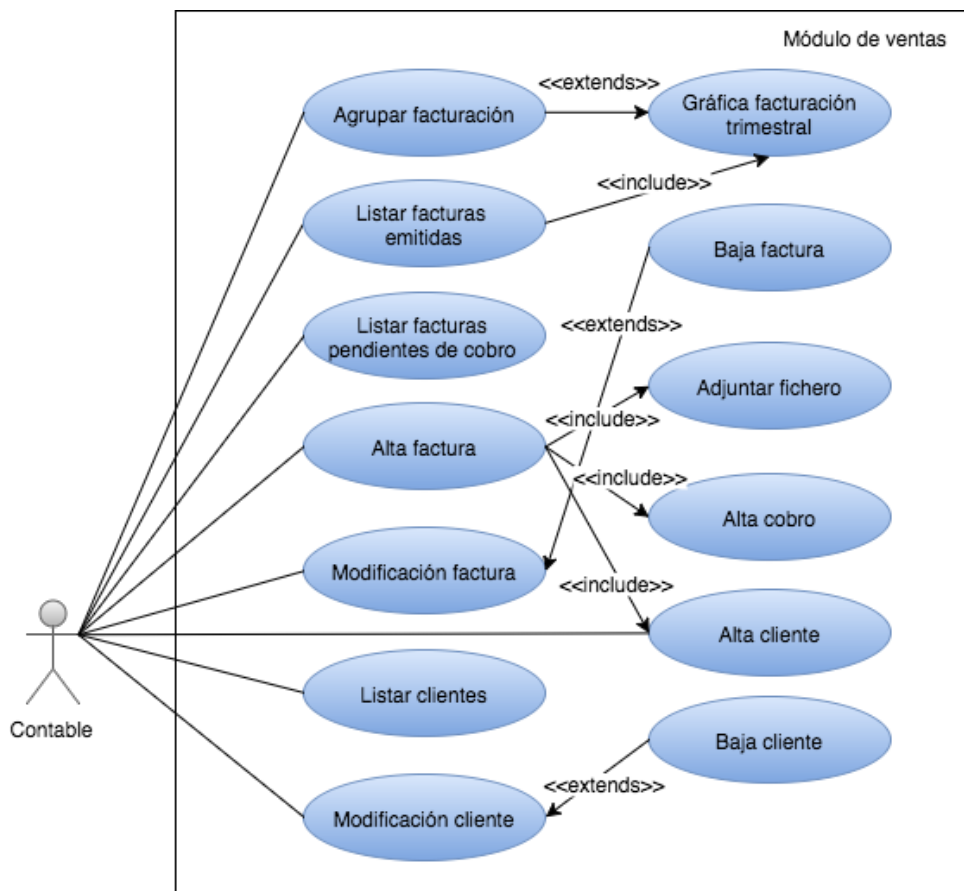


Figura 3.10: Diagramas de casos de uso: Módulo de ventas

Para el escenario del módulo de ventas se han decidido detallar los casos de uso de “Agrupar facturación” y “Listar facturas pendientes de cobro”.

El caso de uso “Agrupar facturación” es un claro caso de relación de especialización del caso de uso “Gráfica facturación trimestral”, tal como muestra la figura 3.10.

<i>Identificador</i>	UC-9	<i>Objetivo</i>	Agrupar facturación
<i>Escenario</i>	Módulo de ventas	<i>Actores</i>	Contable
<i>Descripción</i>	<p>Modificar la gráfica de facturación trimestral para que muestre la facturación de un año determinado. Los pasos a seguir serán:</p> <ol style="list-style-type: none"> 1. Acceder al módulo de ventas haciendo click en la pestaña <i>ventas</i>. 2. Hacer click en el combo selector “Año” junto a la gráfica “facturación trimestral”. 3. Seleccionar un año. 4. El gráfico mostrará la facturación trimestral del año seleccionado. 		
<i>Pre-condiciones</i>	Debe existir al menos una factura creada en el sistema para poder agrupar la gráfica.		
<i>Post-condiciones</i>	La gráfica de facturación trimestral debe mostrar la facturación del año seleccionado.		

Tabla 3.86: UC-9. Agrupar facturación

El caso de uso “Agrupar facturación” es un claro caso de relación de especialización del caso de uso “Gráfica facturación trimestral”, tal como muestra la figura 3.10.

<i>Identificador</i>	UC-10	<i>Objetivo</i>	Listar facturas pendientes
<i>Escenario</i>	Módulo de ventas	<i>Actores</i>	Contable
<i>Descripción</i>	<p>Listar las facturas pendientes de cobro. Los pasos a seguir serán:</p> <ol style="list-style-type: none"> 1. Acceder al módulo de ventas haciendo click en la pestaña <i>ventas</i>. 2. La gráfica aparecerá en la esquina superior derecha del módulo de ventas. 		
<i>Pre-condiciones</i>	Debe existir al menos una factura que no tenga fecha de cobro y su fecha de vencimiento sea anterior a la fecha actual.		
<i>Post-condiciones</i>	El listado mostrará el detalle del tiempo que lleva pendiente por pagar la factura.		

Tabla 3.87: UC-10. Listar facturas pendientes

3.3.6. Casos de uso del módulo de proyectos

Si se explota el caso de uso *proyectos* del caso de uso general, observamos los casos de uso del módulo de proyectos detallados a continuación.

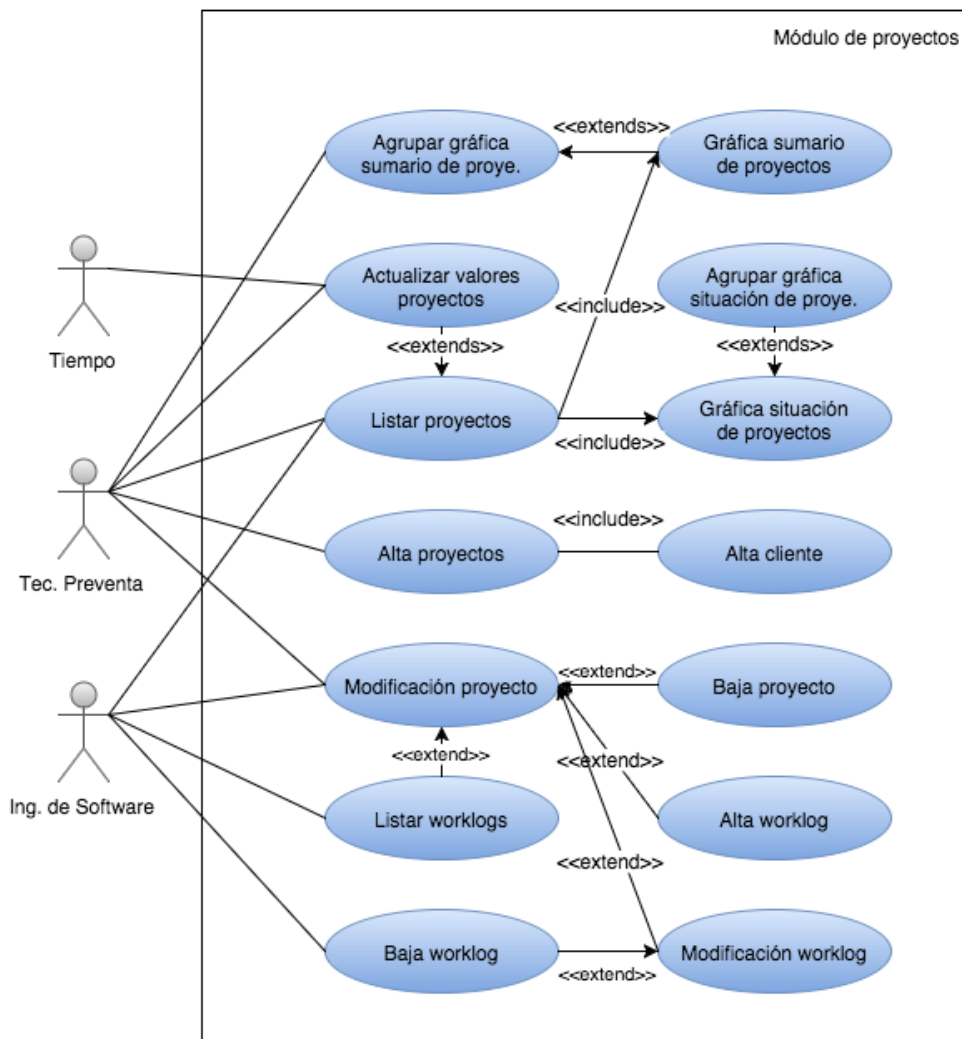


Figura 3.11: Diagramas de casos de uso: Módulo de proyectos

Para el escenario del módulo de proyectos, se han decidido detallar los casos de uso de “Actualizar valores” y “Alta worklog”.

El caso de uso “Actualizar valores proyectos” es un caso especial, que puede ser realizado tanto por el actor Técnico preventa, como por el actor tiempo, tal como muestra la figura 3.11.

<i>Identificador</i>	UC-11	<i>Objetivo</i>	Actualizar valores proyectos
<i>Escenario</i>	Módulo de proyectos	<i>Actores</i>	Tec. preventiva, Tiempo
<i>Descripción</i>	<p>Actualizar los valores numéricos de los proyectos. Los pasos a seguir serán:</p> <p>a. Si se trata del actor Tec. preventiva:</p> <ol style="list-style-type: none"> 1. Acceder al módulo de proyectos haciendo click en la pestaña <i>proyectos</i>. 2. Hacer click en el enlace “Actualizar valores” del “listado de proyectos”. 3. Los proyectos del listado mostrarán los nuevos datos calculados. <p>b. Si se trata del actor Tiempo:</p> <ol style="list-style-type: none"> 1. Esperar 5min. 2. Calcular los valores de todos los proyectos y actualizarlos. 3. Volver al paso 1. 		
<i>Pre-condiciones</i>	Debe existir al menos un proyecto creado en el sistema.		
<i>Post-condiciones</i>	Los proyectos del listado deben mostrar los últimos valores calculados.		

Tabla 3.88: UC-11. Actualizar valores

<i>Identificador</i>	UC-12	<i>Objetivo</i>	Alta worklog
<i>Escenario</i>	Módulo de proyectos	<i>Actores</i>	Ing. de software
<i>Descripción</i>	<p>Crear un reporte de trabajo de un empleado. Los pasos a seguir serán:</p> <ol style="list-style-type: none"> 1. Acceder al módulo de proyectos haciendo click en la pestaña <i>proyectos</i>. 2. Hacer click en el nombre del proyecto en el que se quiera hacer el reporte. 3. Hacer click en el botón <i>añadir</i> bajo el listado de <i>últimos worklogs</i>. 4. Rellenar los datos de clave, fecha y tiempo. 5. Seleccionar el empleado del listado de <i>técnicos</i>. 6. Hacer click en el botón <i>guardar</i>. 		
<i>Pre-condiciones</i>	Debe existir al menos un proyecto creado en el sistema para poder crear un worklog. Debe existir al menos un empleado activo en el sistema para poder crear un worklog.		
<i>Post-condiciones</i>	El listado de últimos worklogs deberá mostrar el nuevo worklog creado.		

Tabla 3.89: UC-12. Alta worklog

3.3.7. Casos de uso del módulo de balance

Si se explota el caso de uso *balance* del caso de uso general, observamos los casos de uso del módulo de balance detallados a continuación.

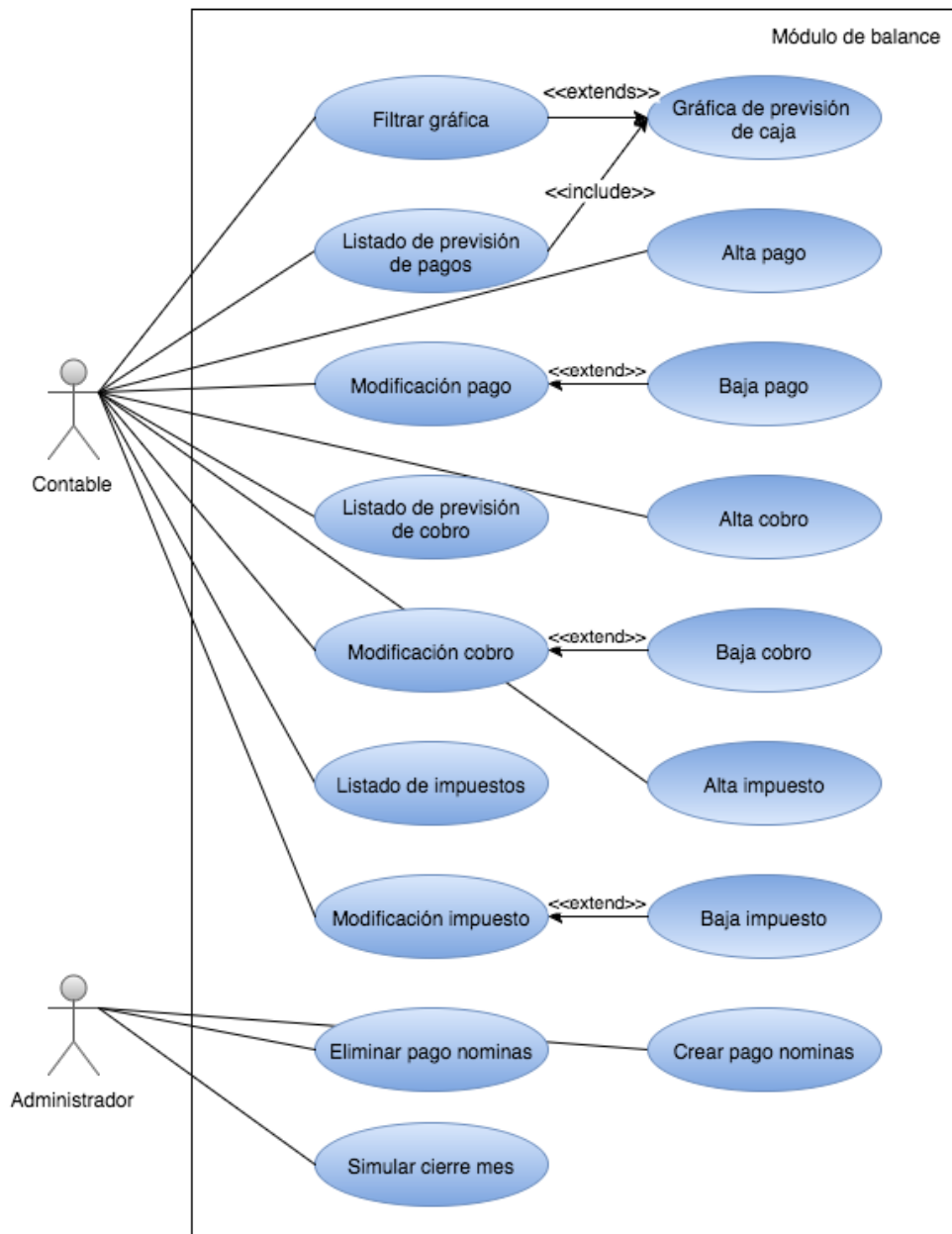


Figura 3.12: Diagramas de casos de uso: Módulo de balance

Para el escenario del módulo de balance, se han decidido detallar los casos de uso de “Modificar pago” y “Alta Impuesto”.

<i>Identificador</i>	UC-13	<i>Objetivo</i>	Modificar pago
<i>Escenario</i>	Módulo de balance	<i>Actores</i>	Contable
<i>Descripción</i>	<p>Actualizar la fecha o el importe de un pago previsto por la empresa. Los pasos a seguir serán:</p> <ol style="list-style-type: none"> 1. Acceder al módulo de balance haciendo click en la pestaña <i>balance</i>. 2. Hacer click en la fecha del pago desde el listado “previsión de pagos”. 3. Modificar los datos del pago y hacer click en guardar. 		
<i>Pre-condiciones</i>	<p>Debe existir al menos un pago en el sistema.</p> <p>Los datos del pago deben ser válidos al guardar.</p>		
<i>Post-condiciones</i>	<p>El gráfico de previsión mostrará el pago en la nueva fecha de haberse cambiado.</p> <p>El listado de previsión de pagos mostrará el pago con los nuevos datos.</p>		

Tabla 3.90: UC-13. Modificar pago

<i>Identificador</i>	UC-14	<i>Objetivo</i>	Alta impuesto
<i>Escenario</i>	Módulo de balance	<i>Actores</i>	Contable
<i>Descripción</i>	<p>Crear un nuevo pago de impuesto programado para un trimestre de un año determinado. Los pasos a seguir serán:</p> <ol style="list-style-type: none"> 1. Acceder al módulo de balance haciendo click en la pestaña <i>balance</i>. 2. Hacer click en el botón <i>añadir</i> bajo el listado <i>previsión de impuestos</i>. 3. Seleccionar el ejercicio y trimestre en el que se debe hacer efectivo el pago del impuesto. 4. Hacer click en guardar. 		
<i>Pre-condiciones</i>	<p>Deben estar correctamente definidas las fechas de los distintos trimestres impositivos en el <i>módulo de configuración</i>.</p> <p>Los datos del impuesto deben ser válidos al guardar.</p>		
<i>Post-condiciones</i>	<p>El sistema creará una transacción de pago asociada al impuesto, y coincidiendo con la fecha de éste, de manera automática. El gráfico de previsión mostrará el pago del impuesto en la fecha que tenga configurada.</p> <p>El listado de previsión de pagos mostrará el pago del impuesto con los datos aportados.</p>		

Tabla 3.91: UC-14. Alta impuesto

3.3.8. Casos de uso del sistema

El *caso de uso general* no describe una funcionalidad, que resulta primordial para los objetivos que se persiguen con el desarrollo de la aplicación: la gestión automática del proceso de cambio de mes para todo los elementos del sistema. Los casos de uso internos de este proceso se pueden observar en la figura 3.13.

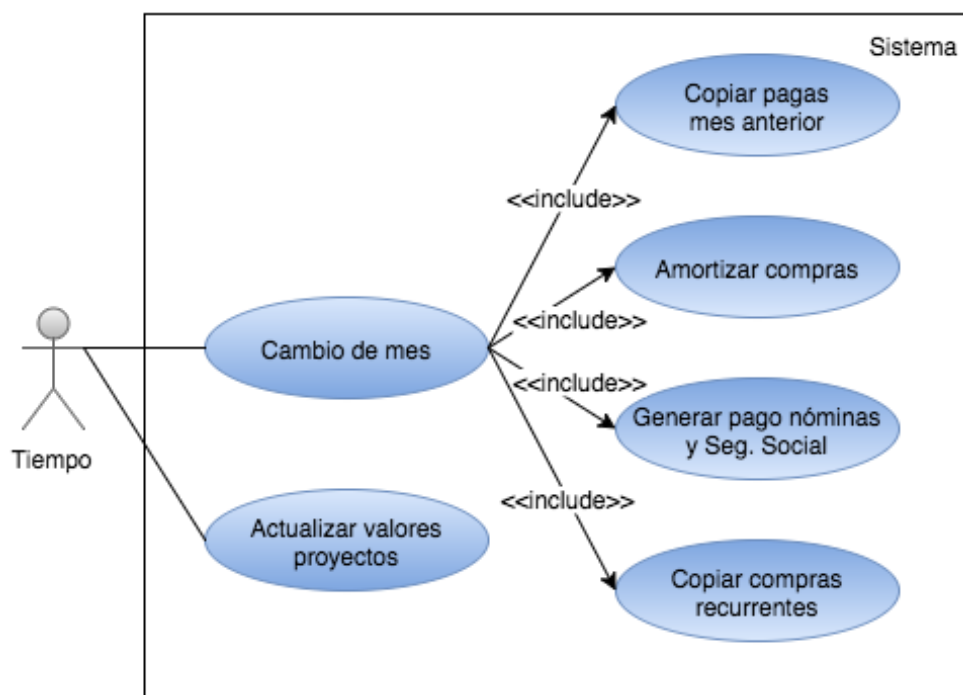


Figura 3.13: Diagramas de casos de uso: Sistema

Identificador	UC-15	Objetivo	Cambio de mes
Escenario	Sistema	Actores	Tiempo
Descripción	Preparar la contabilidad de la aplicación, para el cambio de mes. Los pasos a seguir serán: 1. Copiar las últimas pagas de los empleados activos. 2. Amortizar las compras que no estén completamente amortizadas. 3. Generar transacciones de pago de nóminas y SS. 4. Copiar las operaciones de compra marcadas como “se repite”.		
Pre-condiciones	La fecha de sistema debe coincidir con el día 1 del mes a las 00:00		
Post-condiciones			

Tabla 3.92: UC-15. Cambio de mes

Capítulo 4

Diseño

En este capítulo, se profundiza en la arquitectura de software elegida para esta aplicación, y se detallan los componentes de esta arquitectura: se presenta el modelo de datos y el diseño de la interfaz de usuario.

4.1. Arquitectura de Software

En esta sección se describe la arquitectura de software elegida para desarrollar la herramienta solicitada. Como hemos podido ver en el capítulo anterior, uno de los requisitos a cumplimentar, es la disponibilidad de acceso a la herramienta a través de internet.

Debido a esto, se tomó la decisión de diseñar la aplicación utilizando una arquitectura de desarrollo web siguiendo el *modelo cliente-servidor*. Particularmente, utilizando el patrón de diseño web *modelo-vista-controlador*.

4.1.1. Modelo Cliente - Servidor

Las aplicaciones web, tal como las conocemos, se basan en una arquitectura *cliente-servidor*.

El modelo cliente-servidor, es un modelo de aplicación distribuida, en el que las acciones se ejecutan en uno o varios proveedores de recursos o servicios, llamados *servidores* y son invocadas desde un consumidor de servicios o *cliente*.

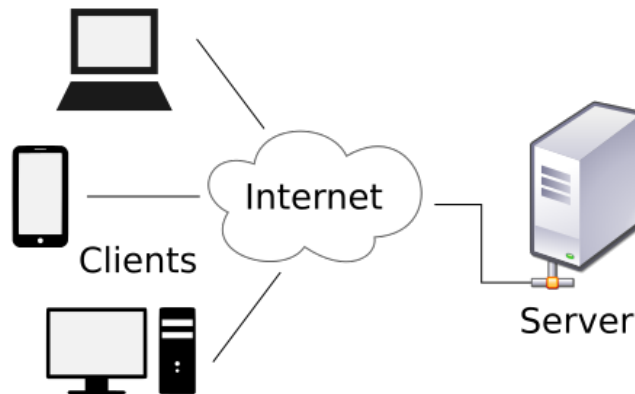


Figura 4.1: Modelo cliente-servidor

El flujo de esta arquitectura es:

1. El cliente envía una petición al servidor.
2. El servidor de aplicaciones, identifica el objetivo de la petición, si puede, la ejecuta y envía los datos de vuelta al cliente. Si no puede, informa del error.
3. El cliente interpreta los datos recibidos y los muestra por la interfaz de usuario.

4.1.2. Modelo-Vista-Controlador

Se denomina modelo-vista-controlador, a un patrón de diseño web, cuya principal característica es, la separación en tres capas de los componentes de una aplicación web: los datos de la aplicación (modelo), la interfaz de usuario (vista) y la lógica de control (controlador).

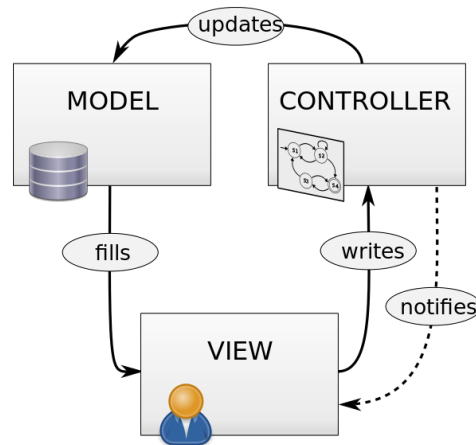


Figura 4.2: Patrón Modelo-Vista-Controlador

Modelo

Contiene el núcleo de la funcionalidad (dominio) de la aplicación.

Encapsula el estado de la aplicación.

Es independiente de la vista y el controlador.

Vista

Es la presentación del Modelo.

Puede acceder al modelo, pero nunca modificar su estado.

Puede ser notificada cuando hay un cambio de estado en el modelo.

Controlador

Reacciona a la petición del cliente, ejecutando la acción adecuada y modificando el modelo convenientemente.

Grails

El framework de desarrollo Grails, aporta una solución *out-of-the-box* para implementar el patrón modelo-vista-controlador, mediante tres herramientas, basadas en su principio “CoC” [8] (Convention over configuration).

- Controllers.

Cualquier clase groovy, dentro de la carpeta */controllers* en la raíz del proyecto, que siga la notación **Controller.groovy* (es decir, finalice con la palabra *Controller*) se convertirá automáticamente en un controlador, capaz de aceptar peticiones en un recurso web determinado por su nombre, y ejecutarlas contra el modelo.

El siguiente ejemplo ilustra un controlador básico, con una sola acción (método), que busca a una persona en la base de datos por su *dni* y lo envía a la vista para mostrarlo. Si no lo encuentra, devuelve un mensaje de error.

```
class PersonaController {  
    def show = { String dni ->  
        def persona = Persona.findByDni(dni)  
        if (persona) {  
            [model: ['persona': persona]]  
        } else {  
            [model: ["errorMessage": "persona no encontrada"]]  
        }  
    }  
}
```

Figura 4.3: Grails MVC: */grails-app/controllers/PersonaController.groovy*

La ruta del recurso web generado por éste controlador y su acción seguirá la misma nomenclatura que su nombre y el de su método: `http://**/persona/show`

- Domain.

Cualquier clase groovy, dentro de la carpeta */domain* en la raíz del proyecto, será considerada una clase del modelo de la aplicación, y como tal esto producirá un mapeo automático de la clase con una tabla de la base de datos, convirtiendo cada propiedad de la clase en una columna de la tabla, con el tipo de datos adecuado según el tipo declarado en la propiedad.

```
class Persona {
    String nombre
    String apellidos
    String dni
    Integer edad
}
```

Figura 4.4: Grails MVC: /grails-app/domain/Persona.groovy

Así mismo, de manera automática nuevos métodos dinámicos serán inyectados en la clase tales como: *save*, *delete*, *findBy**, *countBy** que automatizan la tarea de acceso y modificación del modelo (el método dinámico *findByDni* utilizado en el ejemplo anterior del controlador, no existe realmente en la declaración de la clase *Persona* como vemos, sin embargo, podemos hacer uso de él desde el controlador, al tratarse de un código que grails inyecta de manera dinámica en la clase para nosotros).

- Views.

Las vistas cierran el círculo que relaciona el modelo con el controlador. Para cada controlador definido en la aplicación, grails esperará una carpeta con la misma nomenclatura dentro de la carpeta /views de la raíz del proyecto. Si el nombre de algún fichero de esa carpeta, coincide con el nombre de algún método del controlador, ésta se convertirá en la vista por defecto que se devolverá tras la ejecución del método del controlador.

```
<%@ page contentType="text/html; charset=UTF-8" %>
<html>
<head>
    <title>Detalle persona</title>
</head>
<body>

    <h1>Persona</h1>

    <h2>Datos personales:</h2>

    <span>Nombre:</span> <span>${persona.nombre}</span>
    <span>Apellidos:</span> <span>${persona.apellidos}</span>
    <span>DNI:</span> <span>${persona.dni}</span>
    <span>Edad:</span> <span>${persona.edad}</span>

</body>
</html>
```

Figura 4.5: Grails MVC: /grails-app/views/persona/show.groovy

La última línea del método será el valor devuelto, que se considerará el “modelo” que se utilizará para construir la vista.

4.2. Diseño del Modelo de datos

En esta sección se describe el diseño de la base de datos de la aplicación, mediante el diagrama de Entidad-Relación y el modelo relacional de la base de datos.

4.2.1. Diagrama de Entidad-Relación

Un diagrama de entidad-relación, también denominado *modelo entidad-relación*, es una herramienta para el modelado de datos que permite representar las entidades del modelo, así como sus interrelaciones y propiedades.

Para esta representación gráfica se utilizarán los siguientes elementos:

- Entidad: Simboliza un objeto del mundo real o un concepto que se pretende representar. Las características de cada entidad, se representan mediante *atributos* y *claves*. Las claves primarias se representan con una bola negra. Las claves foráneas se representan con una bola gris. Los atributos se representan con una bola blanca. Su representación se muestra en la figura 4.6:

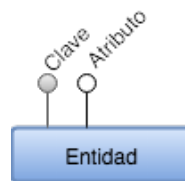


Figura 4.6: Diagrama E/R: Entidad

- Relación: Simboliza la relación entre dos entidades del modelo. Esta relación, según su naturaleza, podrá ser clasificada como: dependencia, herencia o participación. Su representación se muestra en la figura 4.7:

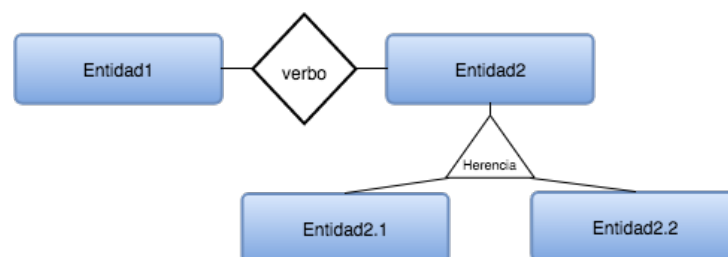


Figura 4.7: Diagrama E/R: Relación

- Restricción: Simboliza la cardinalidad de las relaciones entre las entidades, indica el numero de entidades con las que puede estar relacionada una entidad dada. Su representación se muestra en la figura 4.8:

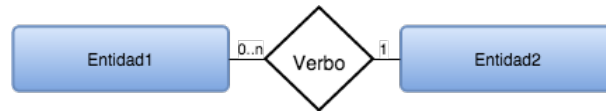


Figura 4.8: Diagrama E/R: Restricción

Para evitar duplicidad de información, sólo se van a mostrar las claves de las entidades que aporten significado y relevancia al diagrama, ya que el *modelo relacional* contiene todos los atributos de cada entidad.

Diagrama E/R: General

El siguiente diagrama muestra las entidades, relaciones y restricciones de todos los elementos del sistema.

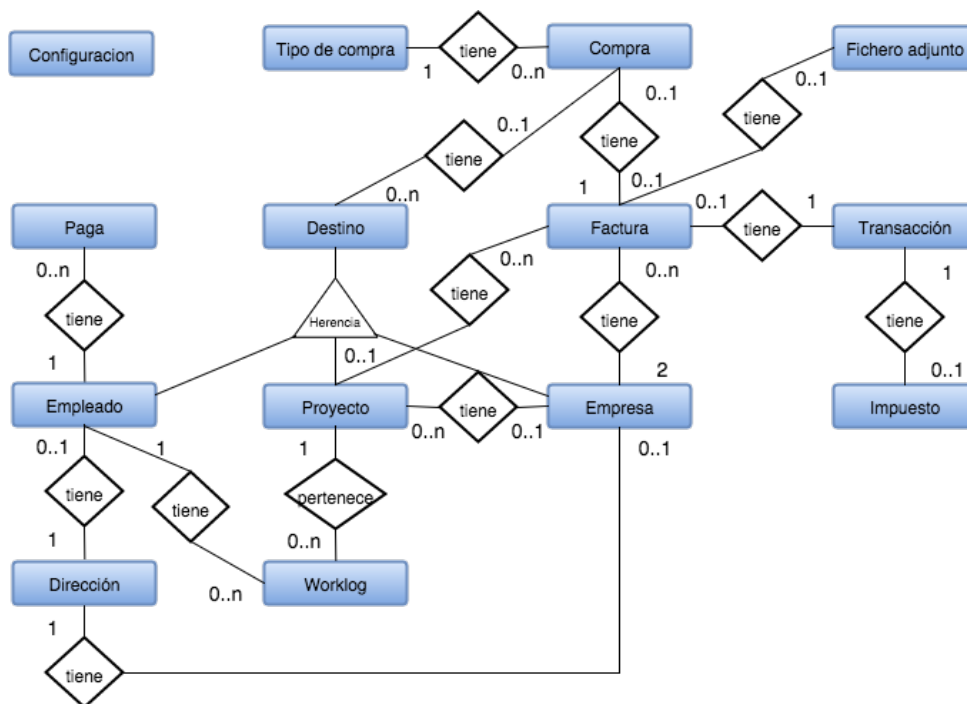


Figura 4.9: Diagrama E/R: General

Diagrama E/R: Empleado

El siguiente diagrama muestra las entidades, relaciones y restricciones de los empleados.

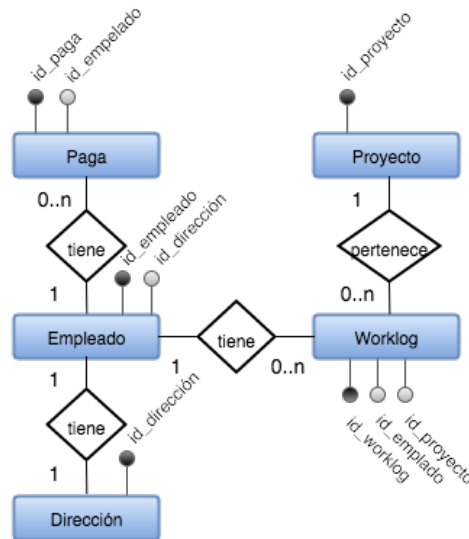


Figura 4.10: Diagrama E/R: Empleados

Diagrama E/R: Compras

El siguiente diagrama muestra las entidades, relaciones y restricciones de las compras.

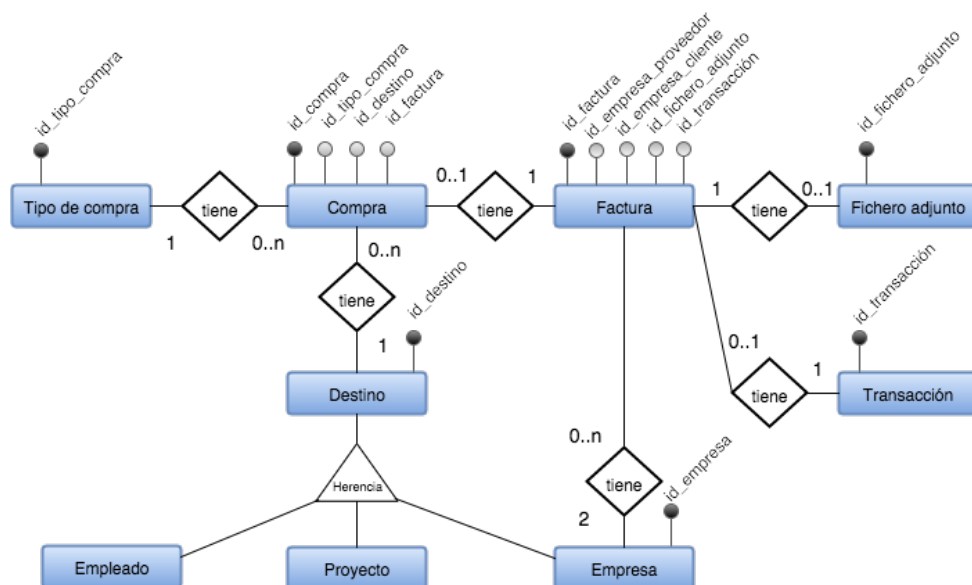


Figura 4.11: Diagrama E/R: Compras

Diagrama E/R: Ventas

El siguiente diagrama muestra las entidades, relaciones y restricciones de las ventas.

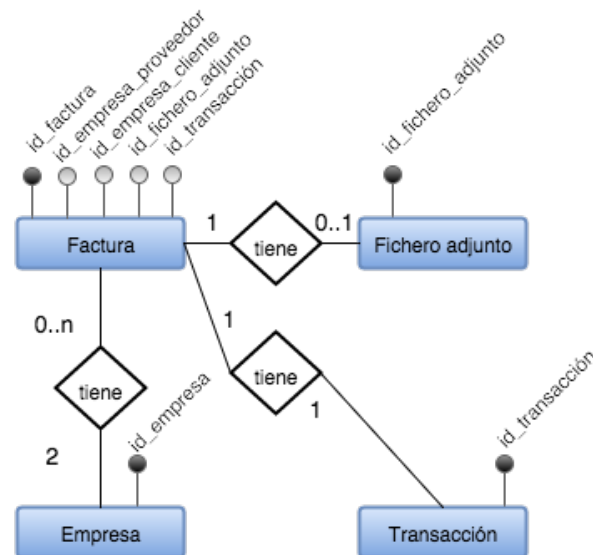


Figura 4.12: Diagrama E/R: Ventas

Diagrama E/R: Proyectos

El siguiente diagrama muestra las entidades, relaciones y restricciones de los proyectos.

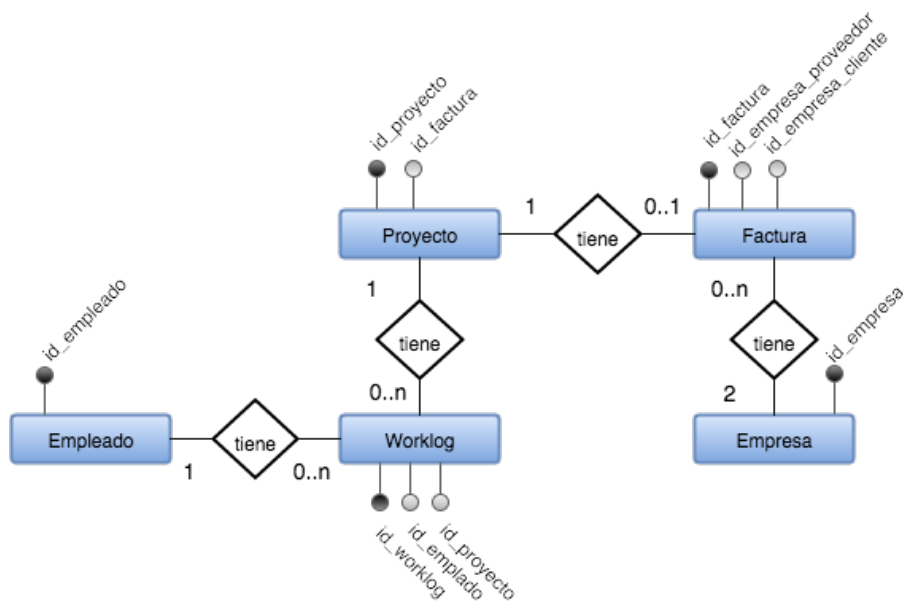


Figura 4.13: Diagrama E/R: Proyectos

Diagrama E/R: Balance

El siguiente diagrama muestra las entidades, relaciones y restricciones del balance.

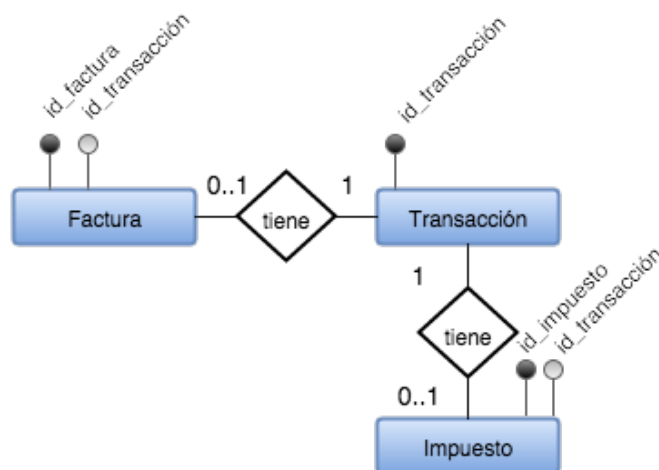


Figura 4.14: Diagrama E/R: Balance

4.2.2. Modelo relacional

El modelo relacional, es la representación gráfica de un conjunto de tablas de una base de datos, en los que los campos de unas tablas se relacionan con otras, por algún campo.

Este campo se conoce como clave foránea, y comparte el mismo tipo y longitud en ambas tablas.

La siguiente figura 4.15 muestra el modelo relacional de la aplicación pgs:

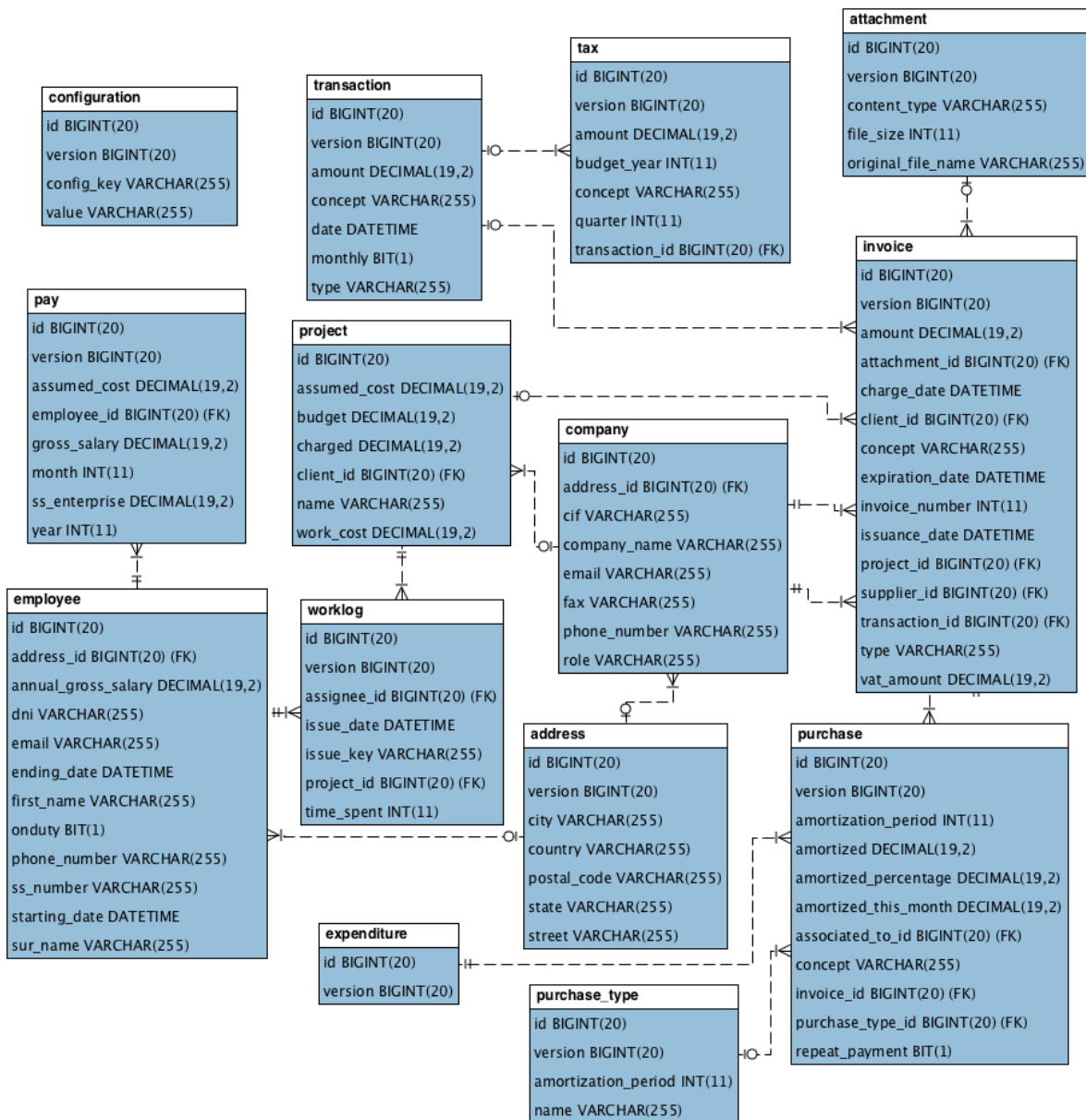


Figura 4.15: Modelo Relacional

4.3. Diseño de la Interfaz de usuario

En esta sección se describe el diseño de la interfaz de usuario elegida, siguiendo las indicaciones de los requisitos: interfaz sencilla, de rápida respuesta, con ventanas que se muestren y oculten de manera automática.

Siguiendo estas indicaciones, se presentó una primera propuesta o boceto de la aplicación, que fue modificada por la empresa solicitante del desarrollo, para llegar finalmente al siguiente diseño:

4.3.1. Módulo de empleados

A continuación se presentan los diseños de la pestaña de empleados.

Módulo de empleados

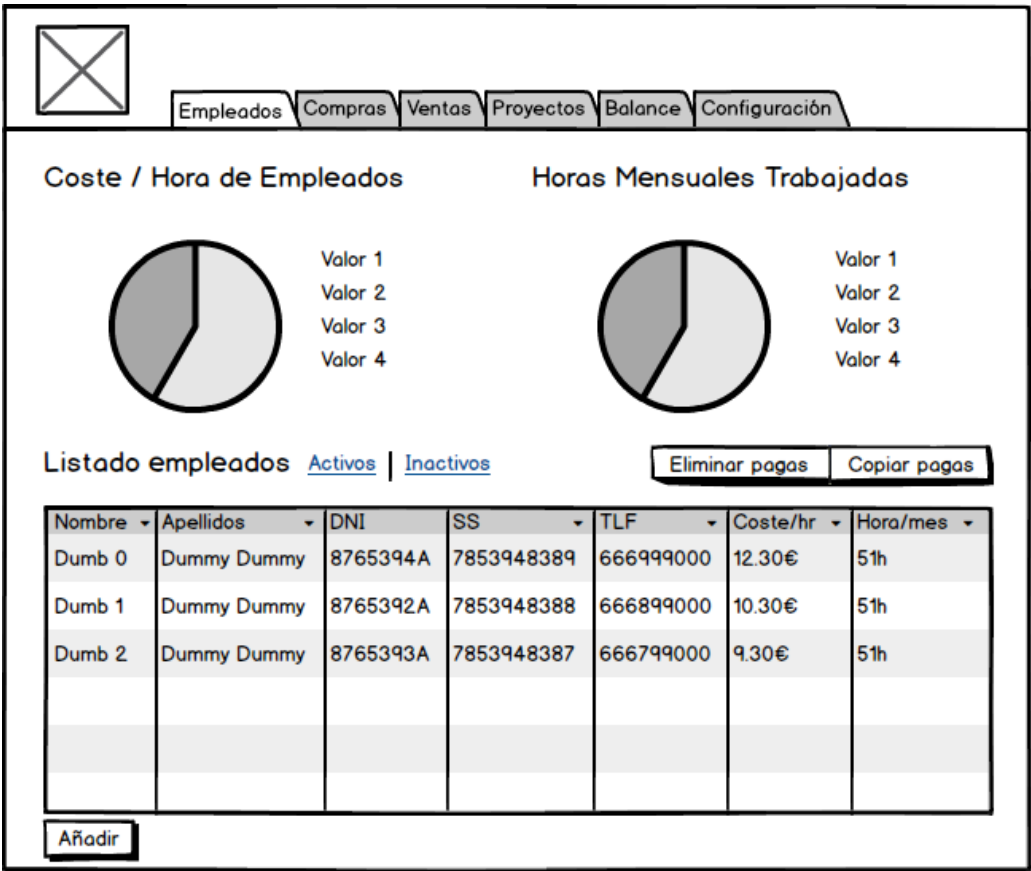


Figura 4.16: Diseños: Pestaña empleados

Detalle listado bajas

Empleados Compras Ventas Proyectos Balance Configuración

Coste / Hora de Empleados Horas Mensuales Trabajadas

Valor 1
Valor 2
Valor 3
Valor 4

No hay datos

Listado empleados [Activos](#) | [Inactivos](#) [Herramientas Admin](#)

Nombre	Apellidos	DNI	SS	TLF	Coste/hr	Hora/mes	Estado
Dumb 0	Dummy Dummy	8765394A	7853948389	666999000	12.30€	51h	Baja

Añadir

Figura 4.17: Diseños: Detalle baja empleado

Detalle de empleado

Detalles de Empleado

Información Personal

Nombre: Dummy Apellidos: Dummy Dummy

DNI: 11222345A Teléfono: 666999111

Email: dummy@dummy.com

Dirección

Domicilio: dummy, 13

Ciudad: Dummy Provincia: Dummy

CP: 11222345A País: Dummy

Información Laboral

Salario Bruto Anual: #####

Fecha de alta: dd/mm/yyyy

Número SS: 111122223333

Me	Af	Salario Br	Coste Asur	Coste Tr
3	2014	1345€	0€	1455€
2	2014	1345€	0€	1455€
1	2014	1345€	100€	1125€

Añadir

Cancelar Baja Actualizar

Figura 4.18: Diseños: Detalle empleado

Detalle paga mensual empleado

Detalles de Paga Mensual

Paga Mensual

Salario Bruto

####€

SS Empresa

####€

Coste Asumido

####€

Coste Total

####€

Mes

Abril

Año

2014

Cancelar

Baja

Actualizar

Figura 4.19: Diseños: Detalle paga mensual empleado

4.3.2. Módulo de compras

A continuación se presentan los diseños de la pestaña de compras.

Módulo de empleados

Sumario de Compras Prov ▼

Valor 1
Valor 2
Valor 3
Valor 4

Listado de Proveedores

Razón Social	Email	Teléfono
Dumb 0	d@ummy.com	897867887
Dumb 1	d@ummy.com	897867887
Dumb 2	d@ummy.com	897867887

Últimas Compras [Herramientas Admin](#)

Gestionar tipos de compra Des-amortizar Compras Amortizar Compras

Fecha	Concepto	Proveedor	Importe	Amortizado
11/11/2014	Test	Proveedor1	####€	####€
11/07/2014	Test	Proveedor2	####€	####€
11/05/2014	Test	Proveedor3	####€	####€

Añadir

Figura 4.20: Diseños: Pestaña compras

Detalle de compra

Detalles de Compra

Datos de Compra

Concepto

Dummy

Asociado a

Dummy

Tipo

Software

Periodo Amort.

48

Amortizado

####€

Datos de Compra

Proveedor

Dummy

Compra

dd/mm/yyyy

Cliente

Dummy

Pago

dd/mm/yyyy

Importe

####€

IVA

####€

☐ Repetición mensual

Archivo Adjunto

File

Cancelar

Eliminar

Actualizar

Figura 4.21: Diseños: Detalle de compra

Detalle de proveedor

Detalles de Proveedor

Información de la Empresa

Razón Social

Dummy

Teléfono

666999222

CIF

A-11222345

Fax

666999111

Email

dummy@dummy.com

Dirección

Domicilio

dummy, 13

Ciudad

Dummy

Provincia

Dummy

CP

11222345A

País

Dummy

Cancelar

Eliminar


Actualizar

Figura 4.22: Diseños: Detalle de proveedor

4.3.3. Módulo de ventas

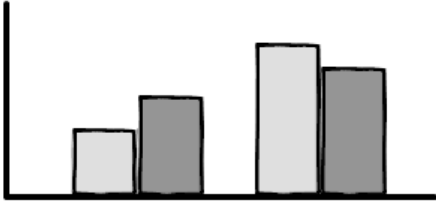
A continuación se presentan los diseños de la pestaña de ventas.

Módulo de ventas



Empleados
Compras
Ventas
Proyectos
Balance
Configuración

Facturación Trimestral
2014 ▼



Facturas Pendientes de Cobro

Factura	Fecha	Retraso	Cliente	Importe
Dumb0	11/12/2014	2m	Dummy	####€
Dumb0	11/12/2014	2m	Dummy	####€
Dumb0	11/12/2014	2m	Dummy	####€

Facturas Emitidas

Factura	Emisión	Cliente	Importe
001	12/12/2014	Dumb0	####€
012	12/04/2014	Dumb0	####€
003	12/03/2014	Dumb0	####€

Listado de Clientes

Razón Social	Email	Teléfono	Pago Medio
Dumb0	d@dummy.com	666111999	0
Dumb0	d@dummy.com	666111999	0
Dumb0	d@dummy.com	666111999	0

Añadir

Añadir

Figura 4.23: Diseños: Pestaña ventas

Detalle de factura

Detalles de Factura

Datos de Facturación

Factura

###

Emisión

dd/mm/yyyy

Concepto

###

Vencimiento

dd/mm/yyyy

Proveedor

Dummy

Cobro

dd/mm/yyyy

Cliente

Dummy

+

Importe

###€

Proyecto

Dummy

IVA

###€

Archivo Adjunto

File

Cancelar

Eliminar

Actualizar

Figura 4.24: Diseños: Detalle de factura

Detalle de cliente

Detalles de Cliente

Información de la Empresa

Razón Social

Dummy

Teléfono

666999222

CIF

A-11222345

Fax

666999111

Email

dummy@dummy.com

Dirección

Domicilio

dummy, 13

Ciudad

Dummy

Provincia

Dummy

CP

11222345A

País

Dummy

Cancelar

Eliminar

Actualizar

Figura 4.25: Diseños: Detalle de cliente

4.3.4. Módulo de proyectos

A continuación se presentan los diseños de la pestaña de proyectos.

Módulo de proyectos

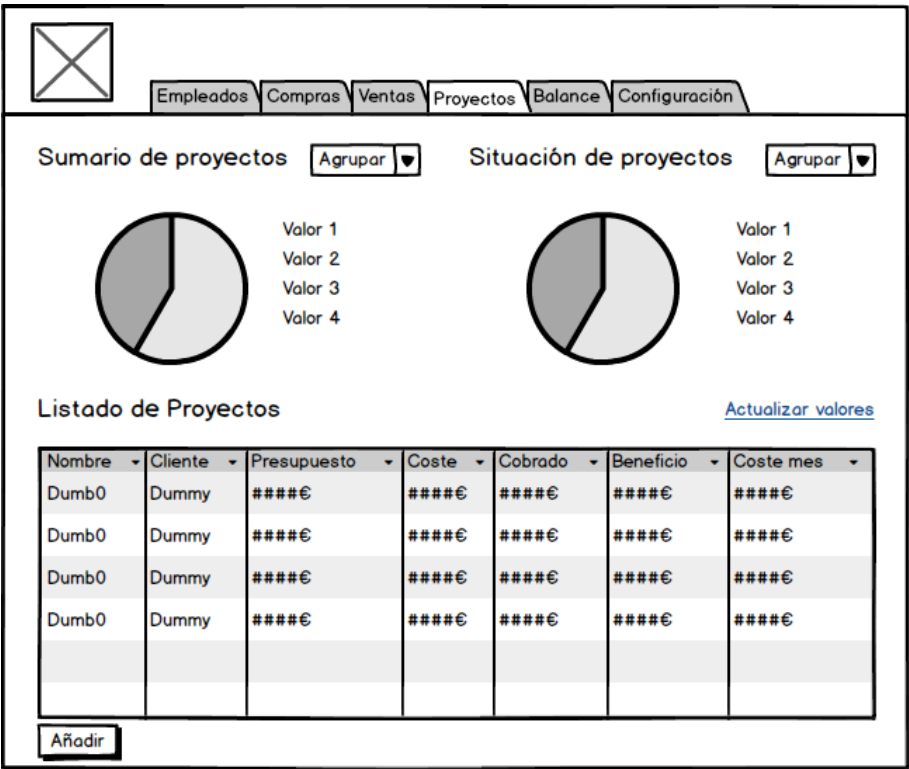


Figura 4.26: Diseños: Módulo de proyectos

Creación de proyecto

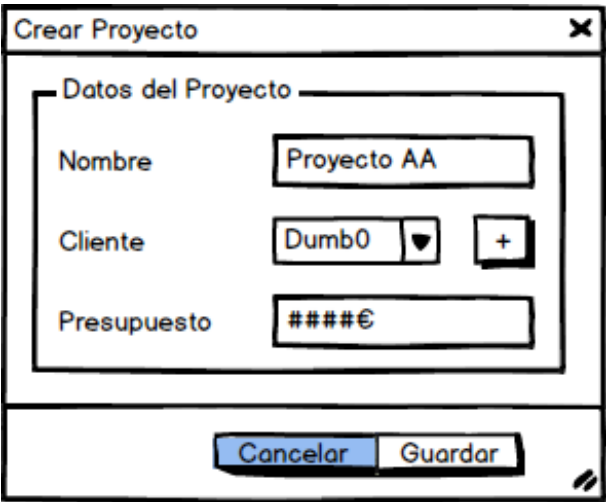


Figura 4.27: Diseños: Creación de proyecto

Detalle de proyecto

Detalles de Factura

Datos de Facturación

Concepto

###

Coste

###€

Cliente

Dummy

+

Cobrado

###€

Proveedor

####€

Beneficio

###€

Ultimos Worklogs

Issue	Fecha	Técnico	Tiempo Empleado
001	11/01/2014	Tec1	1d 5h 23m
001	11/01/2014	Tec1	1d 5h 23m
001	11/01/2014	Tec1	1d 5h 23m

Añadir

Cancelar

Eliminar

Actualizar

Figura 4.28: Diseños: Detalle de proyecto

Detalle de worklog

Detalles del Worklog

Datos del Trabajo Realizado

Issue

Proyecto AA

Fecha

dd/mm/yyyy

Técnico

Dumb0

Tiempo Empleado

#dd #h #m #s

Cancelar

Eliminar

Actualizar

Figura 4.29: Diseños: Detalle de worklog

4.3.5. Módulo de balance


A continuación se presentan los diseños de la pestaña de balance.

Módulo de balance

EmpleadosComprasVentasProyectosBalanceConfiguración

Previsión de Caja

Inicio: / / Fin: / / Filtar



Previsión de pagos

Fecha	Concepto	Importe
dd/mm/yyyy	Dumb0	####€
dd/mm/yyyy	Dumb0	####€
dd/mm/yyyy	Dumb0	####€

Previsión de cobros

Fecha	Concepto	Importe
dd/mm/yyyy	Dumb0	####€
dd/mm/yyyy	Dumb0	####€
dd/mm/yyyy	Dumb0	####€

Previsión de Impuestos

Herramientas Admin

Eliminar pagoCrear pagoSimular cierre mes

Ejercicio	Trimestre	Concepto	Importe	Pago
yyyy	#	Dummy Tax#	####€	dd/mm/yyyy
yyyy	#	Dummy Tax#	####€	dd/mm/yyyy
yyyy	#	Dummy Tax#	####€	dd/mm/yyyy

Añadir

Figura 4.30: Diseños: Módulo de balance

Detalle de pago

Detalles de Pago

Datos de la Operación

Concepto

Dumb0

Fecha

dd/mm/yyyy

Importe

####€

Cancelar

Eliminar

Actualizar

Figura 4.31: Diseños: Detalle de pago

Detalle de cobro

Detalles de Cobro

Datos de la Operación

Concepto

Dumb0

Fecha

dd/mm/yyyy

Importe

####€

Cancelar

Eliminar

Actualizar

Figura 4.32: Diseños: Detalle de cobro

Detalle de impuesto

Detalles de Impuesto

Datos del Impuesto

Concepto

Dumb0

Ejercicio

yyyy

Trimestre

3

Coste Total

####€

Cancelar

Eliminar

Actualizar

Figura 4.33: Diseños: Detalle de impuesto

4.3.6. Módulo de configuración

A continuación se presentan los diseños de la pestaña de configuración.

Módulo de configuración

EmpleadosComprasVentasProyectosBalanceConfiguración

Configuración de la aplicación

Configuración del usuario

Empresa usuario:

Dumb0▼

+

Configuración del usuario

Ruta Ficheros Adjuntos:

//***

Comprobar

Configuración del usuario

Duración semana:

##

 m

Duración jornada:

##

 m

Duración hora:

##

 m

Horas / mes:

##

 h

Actualizar proyectos:

##

 m

Duración semana:

#

▼

yyyy

▼

Duración jornada:

#

▼

yyyy

▼

Duración hora:

#

▼

yyyy

▼

Horas / mes:

#

▼

yyyy

▼

Actualizar proyectos:

dd/mm/yyyy

Reset

Guardar Cambios

Figura 4.34: Diseños: Módulo de configuración

Capítulo 5

Planificación y Presupuesto

En este capítulo, se describe la planificación inicial del proyecto, el diagrama de gantt y el presupuesto junto con el desglose de costes de personal, materiales y totales del desarrollo.

5.1. Planificación del proyecto

La siguiente sección muestra la planificación llevada a cabo para el desarrollo de la aplicación.

La planificación del proyecto se subdividió en historias de usuario y tareas técnicas, siguiendo un flujo de desarrollo iterativo-incremental, basado en Scrum [10].

Cada historia de usuario o *issue* presenta una clave única que la identifica en el proyecto (PGS-XX). Una issue, puede pertenecer o estar englobada en un conjunto de funcionalidad mayor, que se denomina *épica*. Finalmente de cada issue, se presentará una breve descripción o título, la fecha en la que se inició y el tiempo total empleado en la resolución de la misma.

El product backlog del proyecto PGS a la finalización del desarrollo presentaba las siguientes historias de usuario:

<i>Clave</i>	<i>Épica</i>	<i>Descripción</i>	<i>F. Inicio</i>	<i>T. empleado</i>
PGS-1	Viabilidad	Estudio de la situación actual de la empresa y sus necesidades	17/01/11	2d
PGS-2	Viabilidad	Análisis del estado de las tecnologías de desarrollo de aplicaciones web	19/01/11	2d
PGS-3	Análisis	Definición de objetivos primarios y secundarios	21/01/11	1d
PGS-4	Análisis	Toma de requisitos	24/01/11	1d
PGS-5	Análisis	Especificación de la interfaz deseada	25/01/11	1d
PGS-6	Análisis	Planificación y presupuesto	26/01/11	1d
PGS-7	Diseño	Diseño inicial de la interfaz (mockups)	27/01/11	3d
PGS-8	Diseño	Diseño inicial del modelo	01/02/11	2d
PGS-9	Implementa.	Desarrollo del POC basado en Grails	03/02/11	1d
PGS-10	Implementa.	Maquetación de la interfaz y estilos css	04/02/11	6d
PGS-11	Implementa.	Desarrollo del módulo de empleados	14/02/11	10d
PGS-12	Implementa.	Desarrollo del módulo de ventas	28/02/11	10d

PGS-13	Implementa.	Desarrollo del módulo de compras	14/03/11	10d
PGS-14	Implementa.	Desarrollo del módulo de proyectos	28/03/11	10d
PGS-15	Implementa.	Desarrollo del módulo de balance	11/04/11	10d
PGS-16	Implementa.	Desarrollo del módulo de configuración	27/04/11	3d
PGS-17	Implementa.	Desarrollo del proceso cambio mes	03/05/11	3d
PGS-18	Implementa.	Desarrollo de gráficas	06/05/11	4d
PGS-19	Implementa.	Desarrollo de herramientas administrativas	12/05/11	3d
PGS-20	Pruebas	Despliegue y tests manuales	16/05/11	4d
PGS-21	Documenta.	Release Notes + Manual de uso	20/05/11	3d

Tabla 5.1: Planificación del proyecto: Product backlog

5.2. Diagrama de Gantt

La siguiente sección muestra el diagrama de gantt asociado a la planificación de tareas anterior. Para este proyecto se dedicó un recurso (1hombre) a tiempo completo (8h) siguiendo el calendario laboral del año de ejecución del proyecto (2011). Para una mayor claridad visual se han agrupado las tareas dentro de su épica, resultando en la siguiente tabla resumen:

<i>Épica</i>	<i>F. Inicio</i>	<i>T. Empleado</i>	<i>F. Fin</i>
Viabilidad	17/01/11	4d	20/01/11
Análisis	21/01/11	4d	26/01/11
Diseño	27/01/11	5d	02/02/11
Implementación	03/02/11	70d	11/05/11
Pruebas	16/05/11	4d	19/05/11
Documentación	20/05/11	3d	24/05/11

Tabla 5.2: Diagrama de Gantt: PGS

La siguiente figura muestra el diagrama de gantt de la tabla de tareas anterior:

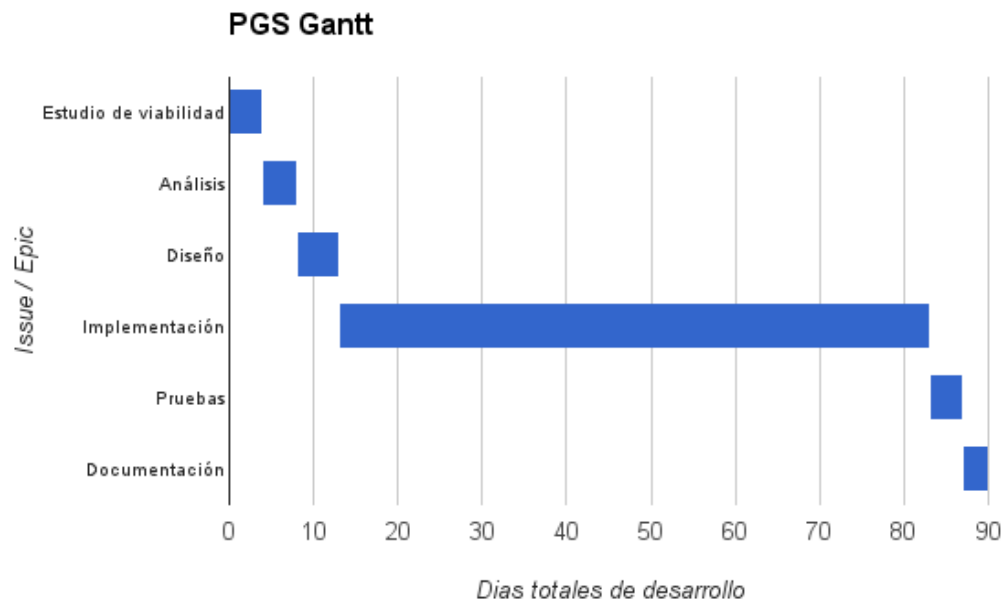


Figura 5.1: Diagrama de Gantt

5.3. Presupuesto

La siguiente sección detalla el presupuesto junto con el desglose de costes de personal, materiales y totales del desarrollo, basandose en las guías propuestas por la Universidad Carlos III de Madrid [18].

5.3.1. Costes de personal

Como se ha comentado en la sección anterior 5.2, los recursos humanos destinados a la elaboración del proyecto fueron:

- 1 hombre con una dedicación del 100 % de la jornada laboral
- jornada laboral fijada en 8h
- semana laboral de 5 días + 2 días de descanso (Sábado y Domingo)
- se realizaron las siguientes festividades:

21 y 22 de Abril de 2011

En base a estos datos, y según la tabla 5.2 de planificación del diagrama de gantt anterior, el total de días dedicados al proyecto se fija en: **90 días**, a contar desde el día 17 de Enero de 2011, hasta el día 26 de Mayo del mismo año, algo más de 4 meses de desarrollo. Esto nos permite calcular los costes de personal:

<i>Recurso</i>	<i>Categoría</i>	<i>Hombres/mes (*)</i>	<i>Coste hombre/mes</i>	<i>Coste (Euros)</i>
Roberto Martín Flores	Ingeniero	4,27	2.694,38 €	11.505,00 €

Tabla 5.3: Presupuesto: Gastos de personal

(*) 1 Hombre mes = 131,25 horas. Máximo anual de dedicación de 12 hombres mes.

El coste de personal se obtiene de la siguiente fórmula:

$$Costedepersonal = \left(\frac{Totaldías * Horas/día}{Dedicacionhombre/mes} \right) * Costehombre/mes$$

Figura 5.2: Fórmula del coste de personal

Los costes totales de personal del proyecto ascienden a ONCE MIL QUINIENTOS CINTO EUROS.

5.3.2. Costes de materiales

Para la realización del proyecto fueron adquiridos ciertos equipos informáticos:

<i>Descripción</i>	<i>Coste (Euro)</i>	<i>% Uso dedicado proyecto</i>	<i>Dedicación (meses)</i>	<i>Periodo de depreciación</i>	<i>Coste Imput.</i>
Ordenador personal	1199,05 €	100	4	60	79,94 €
Servidor	2578,36 €	100	4	60	214,86 €
Total					294,89 €

Tabla 5.4: Presupuesto: Gastos de materiales hardware

^{*)} Fórmula de cálculo de la Amortización:

$$\frac{A}{B} \times C \times D$$

A = nº de meses desde la fecha de facturación en que el equipo es utilizado

B = periodo de depreciación (60 meses)

C = coste del equipo (sin IVA)

D = % del uso que se dedica al proyecto

Figura 5.3: Fórmula del cálculo de la amortización

Los costes totales de materiales ascienden a DOSCIENTOS CUARENTA Y NUEVE CON OCHENTA Y NUEVE CENTIMOS.

5.3.3. Costes de funcionamiento

- Licencia de Jira para 10 usuarios con un coste de 10 €
- Licencia de Microsoft Office 2007 con un coste de 109.90 €
- Licencia de Microsoft Project 2007 con un coste de 129.90 €

Los costes totales de funcionamiento ascienden a DOSCIENTOS CUARENTA Y NUEVE CON OCHENTA CENTIMOS.

5.3.4. Resumen

La siguiente tabla muestra el resumen de costes totales del proyecto:

<i>Concepto</i>	<i>Importe</i>
Personal	11.505,00 €
Amortización	294,89 €
Costes de funcionamiento	294,80 €
Costes indirectos (20 %)	2.418,94 €
Total (sin IVA)	14.513,63 €
Total (IVA 18 %)	17.126,08 €

Tabla 5.5: Presupuesto: Resumen

El presupuesto total de este proyecto asciende a la cantidad de DIECISIETE MIL CIENTO VEINTISÉIS CON OCHO CÉNTIMOS DE EURO.

Leganes, a 26 de Mayo de 2011

El ingeniero proyectista

Fdo. Roberto Martin Flores

Capítulo 6

Implantación y manual de usuario

En este capítulo, se profundiza en el proceso de implantación de la aplicación y se detalla el manual de usuario que permitirá hacer un uso guiado de la aplicación a través de un caso de uso completo.

6.1. Implantación

Esta sección cubre el proceso de instalación de las herramientas web necesarias para desplegar la aplicación pgs en un ordenador con una distribución Linux Debian 5. Los pasos a seguir serán:

- Instalación de Java 6: JRE y JDK
 - o java
 - o java home
- Instalación de Tomcat 5.5
- Instalación de MySQL Server 5.1
 - o usuario + password
 - o bbdd
 - o permisos
- Despliegue

6.1.1. Instalación de Java6: java

Para poder instalar el servidor de aplicaciones Tomcat, necesitamos tener instaladas las librerías java en su versión 6 en la máquina. Para instalar estas librerías deberemos modificar las entradas de nuestro fichero `/etc/apt/sources.list` para poder descargarlas directamente desde los repositorios de Sun:

- deb `http://ftp.us.debian.org/debian/ unstable main contrib non-free`
- deb-src `http://ftp.us.debian.org/debian/ unstable main contrib non-free`
- deb `http://security.debian.org/ testing/updates main contrib non-free`
- deb-src `http://security.debian.org/ testing/updates main contrib non-free`

Hecho esto, ejecutamos la siguientes instrucciones:

- `apt-get update` para actualizar las listas de los repositorios.

- `apt-get install sun-java6-jre sun-java6-jdk` para descargar el JRE y JDK de java 6.

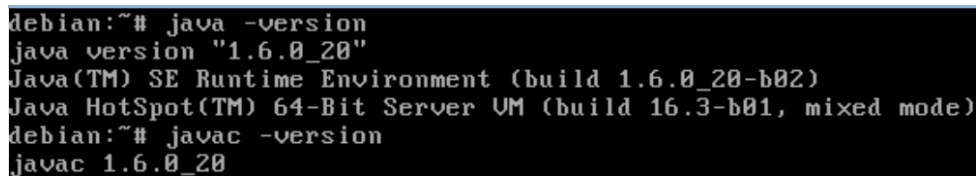
Para comprobar la correcta instalación de las librerías se ejecutan las instrucciones:

- `update-alternatives --config java`
- `update-alternatives --config javac`

y se chequea la versión de java con las instrucciones:

- `java -version`
- `javac -version`

Para saber que se ha instalado correctamente se deberían ver los siguientes mensajes (la versión es la 1.6, pero la subversión, en este caso 0_20, puede variar):



```
debian:~# java -version
java version "1.6.0_20"
Java(TM) SE Runtime Environment (build 1.6.0_20-b02)
Java HotSpot(TM) 64-Bit Server VM (build 16.3-b01, mixed mode)
debian:~# javac -version
javac 1.6.0_20
```

Figura 6.1: Instalación de Java

6.1.2. Instalación de Java6: java home

Para el correcto funcionamiento de las librerías debemos configurar las variables de entorno en nuestra máquina, debemos establecer la variable global `JAVA_HOME` para que las aplicaciones java que ejecutemos encuentren estas librerías. Debemos editar nuestro fichero `/etc/profile` añadiendo:

- `export PATH=$PATH:/usr/lib/jvm/java-6-sun/bin`
- `export JAVA_HOME=/usr/lib/jvm/java-6-sun`
- `export JRE_HOME=/usr/lib/jvm/java-6-sun/jre`

6.1.3. Instalación de Tomcat 5.5

Para poder instalar correctamente Tomcat debemos logarnos como el usuario root. Antes de comenzar la instalación deberemos editar nuestro fichero `/etc/apt/sources.list` añadiendo las siguientes entradas:

- deb <http://ftp.de.debian.org/debian> lenny main contrib
- deb <http://ftp.de.debian.org/debian> lenny main
- Para comenzar la instalación de Tomcat 5.5 hay que ejecutar la siguiente instrucción:
 - o apt-get install tomcat5.5 tomcat5.5-admin tomcat5.5-webapps
- Una vez finalizada la instalación de Tomcat, hay que establecer el valor de la variable `JAVA_HOME` en el fichero `/etc/default/tomcat5.5`:
 - o `JAVA_HOME=/usr/lib/jvm/java-6-sun`

Para comprobar el correcto funcionamiento de Tomcat, deberemos abrir la siguiente url **`http://localhost:8180`** en un navegador web (safari, chrome, ...) y debemos llegar a ver esta página:

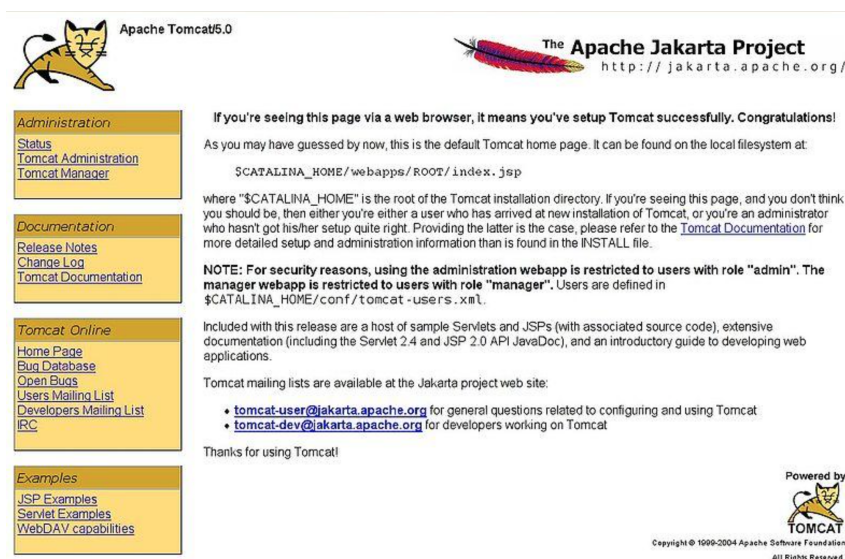


Figura 6.2: Instalación de Tomcat

Para configurar los usuarios que pueden acceder a la administración de Tomcat, hay que acceder y modificar el fichero `$TOMCAT_HOME/etc/tomcat-users.xml` y crear un usuario con los roles “**admin, manager**”.

6.1.4. Instalación de MySQL Server 5.1

Para instalar nuestro servidor de base de datos, deberemos ejecutar la siguiente instrucción:

- `apt-get install mysql-server-5.1`

Durante el proceso de instalación de la base de datos se nos solicitará crear un usuario, con contraseña. Se aconseja crear un usuario **root** con la contraseña deseada para garantizar un mínimo de seguridad.

Un vez creado el usuario **root**, procederemos a crear el usuario necesario para la conexión entre nuestra aplicación y el motor de bases de datos, para ello:

- Hacemos login en la consola de MySQL con el comando: `mysql -u root -p`
- Se nos solicitará la clave. Una vez dentro de la consola interactiva de MySQL, ejecutaremos los siguientes comandos:
 - o `create user 'pgs' identified by 'pgs';`
(Esto creará al usuario pgs, con la password pgs, necesario para la aplicación)
 - o `create database 'pgs' default charset 'utf8';`
(Esto creará la base de datos necesaria para nuestra aplicación)
 - o `grant all privileges on 'pgs'.* to 'pgs'@'localhost';`
(Esto añadirá al usuario pgs los permisos necesarios para nuestra aplicación)

6.1.5. Despliegue

Una vez instalados Tomcat y MySQL, y comprobado que se están ejecutando correctamente, debemos desplegar nuestra aplicación en el servidor. Para ello, simplemente copiaremos el fichero **pgs.war** suministrado en el directorio **webapps** de Tomcat.

Hecho esto deberemos reiniciar nuestro servidor tomcat. Una vez iniciado, encontraremos nuestra aplicación en la url **http://localhost:8080/pgs**

6.2. Manual de usuario

El manual de usuario es una de las partes importantes de este documento, explica cómo utilizar las funcionalidades de la aplicación en detalle desde el punto de vista del usuario según sus necesidades.

6.2.1. Panel de Empleados

La sección dedicada a los empleados de la empresa se presenta en la Figura 6.3 y permite:



Figura 6.3: Panel de empleados

Listado de empleados

El listado de empleados presenta los datos más relevantes de la información de contacto de cada empleado de la empresa, así como su coste/hora para el mes en curso y las horas trabajadas durante el mes en curso. Estos valores vuelven a cero cada primero de mes.

El listado de empleados puede ser ordenado por cualquiera de los campos que forman la tabla, pulsando en el título de cada columna. Cada pulsación invierte el orden de la columna (ascendente, descendente).

En la misma página se puede observar un gráfico comparativo en el que se recogen los 4 empleados con mayor *coste/hora* para el mes en curso y la media del resto. Junto a este, podemos observar otro gráfico comparativo que recoge los 4 empleados con mayor *número de horas trabajadas* para el mes en curso, y la media del resto.

Los empleados de la aplicación pueden estar “activos” o “inactivos” por periodos de tiempo. El estado *inactivo* evitará que se generen automáticamente pagas mensuales para esos empleados, o puedan ser objetivo de alguna compra.

El listado de empleados se muestra filtrado por aquellos empleados en estado *activo* por defecto, aunque es posible también mostrar solo los *inactivos* o *todos* utilizando los links junto al título “Listado de empleados” (*Inactivos*, *Ver todos*).

Creación de empleado

Haciendo click en el botón “Añadir” se despliega el formulario de creación de un empleado tal como muestra la figura 6.4:

The screenshot shows a web application interface for creating a new employee. The top navigation bar includes 'Empleados', 'Compras', 'Ventas', 'Proyectos', 'Balance', and 'Configuración'. The main content area is titled 'Coste / Hora de Empleados' and 'Horas Mensuales Trabajadas'. A red header bar for the form reads 'Crear Empleado'. The form is organized into two columns. The left column, 'Información Personal', contains input fields for 'Nombre', 'Apellidos', 'DNI', 'Teléfono', 'Email', and a 'Dirección' section with sub-fields for 'Domicilio', 'Ciudad', 'Provincia', 'Código Postal', and 'País'. The right column, 'Información Laboral', contains fields for 'Salario Bruto Anual' (displayed as 0,00 €), 'Fecha de alta' (with a calendar icon), and 'Número SS'. At the bottom right of the form, there are two buttons: 'Cancelar' and 'Guardar'.

Figura 6.4: Creación de empleado

Desde él, se puede especificar la información personal del empleado, su dirección, y la información laboral incluido el importe del salario bruto anual.

Edición, baja y eliminación de empleado

Haciendo click en el *nombre* del empleado en el listado de empleados, se despliega el formulario de modificación de un empleado reflejado en la figura 6.5, desde el cual se pueden cambiar sus datos, o darle de baja temporalmente.

Mes	Año	Salario Bruto	Coste Asumido	Coste Total
10	2.015	2.909,50 €	0,00 €	3.084,07 €
9	2.015	2.909,50 €	0,00 €	3.084,07 €
8	2.015	2.909,50 €	4,99 €	2.972,68 €

Nombre	Apellidos	DNI	Número SS	Teléfono	Coste/hora	Horas/mes	Estado
Dumb 6	Dummy Dummy	14886049T	968321830101	91861717	10,99 €	40 h	
Dumb 7	Dummy Dummv	49673014P	999497632324	91062278	16,86 €	55 h	

Figura 6.5: Edición de empleado

Un empleado dado de baja, solo aparecerá por defecto en el listado de empleados *inactivos*, tal como muestra la figura 6.6. Para eliminar el empleado de manera definitiva, deberemos acceder al formulario de modificación de un empleado, que ahora mostrará la opción eliminar. Atención, esta operación no es reversible.



Figura 6.6: Baja de empleado

Listado de pagas mensuales

Desde la pantalla de modificación de empleado presente en la figura 6.5 también se accede al listado de “Últimas Pagas” del empleado.

Últimas Pagas

Mes	Año	Salario Bruto	Coste Asumido	Coste Total
4	2.014	4.166,67 €	507,47 €	4.924,14 €
3	2.014	4.166,67 €	507,47 €	4.924,14 €
2	2.014	4.166,67 €	38,72 €	4.455,39 €

1 2 [Siguiente](#)

 [Añadir](#)

Figura 6.7: Pagas mensuales empleado

Este listado presenta los datos más relevantes de cada paga mensual del empleado: el mes y año de la paga, el salario bruto que percibió el empleado, el coste asumido por amortizaciones y el coste total para la empresa de esa paga.

El listado de pagas puede ser ordenado por cualquiera de los campos que forman la tabla, pulsando en el título de cada columna. Cada pulsación invierte el orden de la columna (ascendente, descendente).

Creación de paga

Haciendo click en el botón *Añadir* se despliega el formulario de creación de una paga mensual tal como muestra la figura 6.8:

Desde él, se puede especificar la información de la paga mensual, su fecha, salario bruto y cuota de la seguridad social. La aplicación calculará automáticamente el coste mensual bruto de la paga dividiendo el salario bruto del empleado entre 12 mensualidades, si lo desea puede modificar este valor.

Análogamente la aplicación calculará el importe de la cuota de la SS calculando el 6% del importe del salario bruto mensual, si lo desea puede modificar este valor. Los campos *coste asumido* y *coste total* son variables calculadas por el sistema y no pueden ser modificadas.

Formulario de creación de una paga mensual. El formulario tiene un título 'Crear Paga Mensual' en un botón rojo. Dentro, se encuentra un subformulario 'Paga Mensual' con los siguientes campos: 'Salario Bruto' (4166,67 €), 'SS Empresa' (250,00 €), 'Coste Asumido' (0,00 €), 'Coste Total' (0,00 €), 'Mes' (Abril) y 'Año' (2014). En la parte inferior del formulario hay dos botones: 'Cancelar' y 'Guardar'.

Figura 6.8: Creación de paga

Edición y eliminación de paga

Desde el listado de pagas, haciendo click sobre el *mes* de una paga, se despliega el formulario de modificación de una paga reflejado en la figura 6.9:

Formulario de edición de una paga mensual. El formulario tiene un título 'Detalles de Paga Mensual' en un botón rojo. Dentro, se encuentra un subformulario 'Paga Mensual' con los siguientes campos: 'Salario Bruto' (4166,67 €), 'SS Empresa' (250,00 €), 'Coste Asumido' (507,47 €), 'Coste Total' (4924,14 €), 'Mes' (Abril) y 'Año' (2014). En la parte inferior del formulario hay tres botones: 'Cancelar', 'Eliminar' y 'Actualizar'.

Figura 6.9: Edición de paga

Atención, la operación de eliminación de una paga no es reversible. Si se elimina o modifica una paga de un mes en curso, se recomienda eliminar las operaciones automáticas de pago de nóminas y cuotas de la SS y volverlas a generar.

Herramientas administrativas

Desde el listado de empleados, haciendo click sobre el enlace “Herramientas administrativas” se muestra el panel de herramientas de usuarios administradores tal como muestra la figura 6.10.

Estas herramientas permiten realizar operaciones de alto riesgo para la aplicación y permiten corregir situaciones anómalas del sistema.

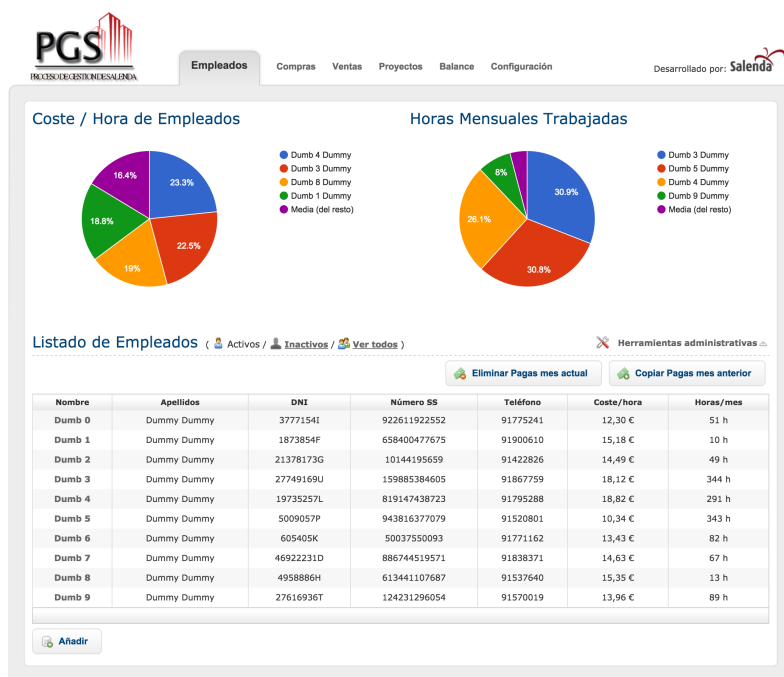


Figura 6.10: Empleados: Herramientas administrativas

Las herramientas administrativas del panel de empleados, permiten:

- **Eliminar pagas mes en curso.** Esta opción eliminará todas las pagas creadas o copiadas del mes anterior por el sistema. Esta opción no es reversible. Tenga en cuenta que esta operación no debe ejecutarse sin antes haber ejecutado la operación “desamortizar compras” de las herramientas administrativas del panel de compras.
- **Copiar pagas mes anterior.** Esta opción emula el proceso automático de cambio de mes, copiando las pagas existentes del mes anterior y asignándoles las fechas del mes en curso. Atención, esta opción tiene un alto riesgo, pues se pueden producir duplicados, se deben haber eliminado las pagas para el mes en curso anteriormente.

6.2.2. Panel de Compras

La sección dedicada a las compras de la empresa se presenta en la Figura 6.11 y permite:



Figura 6.11: Panel de compras

Listado de compras

El listado de compras presenta los datos más relevantes de la información de una compra realizada por la empresa, así como su proveedor, y el porcentaje de amortización acumulado. Este porcentaje se actualizará cada primero de mes.

El listado de compras puede ser ordenado por cualquiera de los campos que forman la tabla, pulsando en el título de cada columna. Cada pulsación invierte el orden de la columna (ascendente, descendente).

En la misma página se puede observar un gráfico comparativo que muestra el porcentaje de *compras agrupadas por proveedor* de los 6 proveedores con más porcentaje y la media del resto.

Junto a este gráfico, encontramos un selector que permite cambiar el criterio de agrupación de la gráfica para observar el porcentaje de *compras agrupadas por el destino o asociación* de esta compra (una compra puede tener como destino o estar asociada a: un

proyecto, un empleado o la propia empresa).

También se pueden *agrupar las compras por su porcentaje de amortización* para observar el porcentaje de amortización completado y pendiente del total gastado en compras, como muestra la figura 6.12.

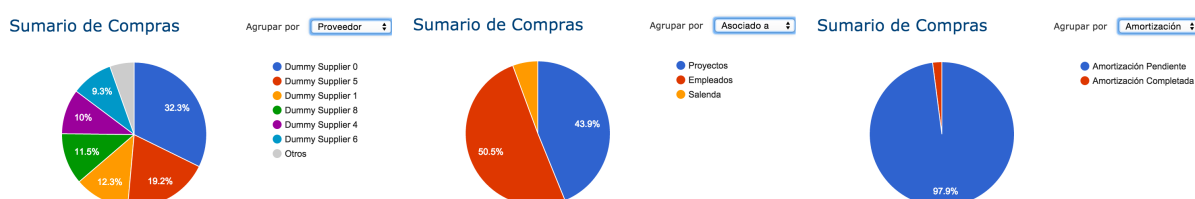


Figura 6.12: Graficas de compras

Creación de compra, proveedor y tipo de compra. Adjuntar fichero

Haciendo click en el botón *Añadir* se despliega el formulario de creación de una compra tal como muestra la figura 6.13.

Una compra debe estar *asociada a* la propia empresa, a un empleado o a un proyecto. Este valor influirá en la manera en la que la compra es amortizada.

También se debe decidir el *tipo de compra* que determinará el periodo (o tiempo) de amortización. Este valor puede ser modificado una vez seleccionado el tipo de compra. También se pueden crear tipos de compra desde el mismo formulario.

Para poder guardar la compra es necesario especificar sus *datos de facturación* tales como: *el proveedor* de la compra (es posible crear un proveedor desde el mismo formulario de creación de compra y seleccionarlo), la *fecha de compra* y la *fecha de pago*, su importe y su cuota de IVA. El cliente de una compra siempre es la empresa de la aplicación por defecto.

Si se trata de una compra recurrente (alquileres, gastos de servicio,...) se puede marcar la opción *Se repite mensualmente*: esto hará que la compra se copie cada primero de mes, modificando las fechas de compra y pago convenientemente, como si se tratara de las pagas de un empleado. Para finalizar la recurrencia de una compra, basta con desactivar la opción en la última compra copiada, esta ya no se duplicará a primero de mes.

Finalmente cada compra puede ir acompañada de un fichero adjunto. No existen restricciones de formato para estos ficheros, sólo se ha de configurar correctamente la ruta donde se alojarán los ficheros en el servidor.

Figura 6.13: Creación de compra

Edición y eliminación de compra

Figura 6.14: Edición de compra

Desde el listado de compras, haciendo click sobre el *concepto* de una compra, se despliega el formulario de modificación de una compra reflejado en la figura 6.14.

Atención: una vez que una compra ha iniciado su amortización, es imposible modificarla (salvo tratarse de una compra recurrente, en cuyo caso se permite desactivar su recurrencia, o adjuntar un fichero diferente. Atención: la operación de eliminación de una compra no es reversible.

Listado de proveedores

El listado de proveedores presenta los datos más relevantes de la información de un proveedor al que la empresa realiza compras asiduamente, como puede observarse en la figura 6.11.

El listado de proveedores puede ser ordenado por cualquiera de los campos que forman la tabla, pulsando en el título de cada columna. Cada pulsación invierte el orden de la columna (ascendente, descendente).

Edición y eliminación de proveedor

Desde el listado de proveedores, haciendo click sobre la *razón social* de un proveedor, se despliega el formulario de modificación de un proveedor reflejado en la figura 6.15.

The screenshot displays the PGS software interface. At the top, there's a navigation bar with tabs: Empleados, Compras, Ventas, Proyectos, Balance, and Configuración. The 'Compras' tab is active. Below the navigation bar, there's a 'Sumario de Compras' (Purchase Summary) section with a pie chart showing percentages: 8.5%, 12.5%, 13.8%, and 22%. To the right of the pie chart is a 'Listado de Proveedores' (Supplier List) table with columns: Razón Social, Email, and Teléfono. The table lists 'Dummy Supplier 0' with email 'we-are@dummyland.com' and phone '91555444'. A modal form titled 'Detalles de Proveedor' is open, showing the 'Información de la Empresa' (Company Information) and 'Dirección' (Address) sections. The 'Información de la Empresa' section includes fields for Razón Social (Dummy Supplier 0), Teléfono (91555444), CIF (B-005864), Fax (91555333), and Email (we-are@dummyland.com). The 'Dirección' section includes fields for Domicilio (Dummy Site, 99), Ciudad (DummyCity), Provincia (Dummystate), Código Postal (22222), and País (DummyLand). At the bottom of the modal, there are buttons for 'Cancelar', 'Eliminar', and 'Actualizar'. The background shows a table of 'Últimas Compras' (Latest Purchases) with columns for Fecha (Date) and Amortizado (Amortized).

Figura 6.15: Edición de proveedor

Atención: la operación de eliminación de un proveedor no es reversible. No es posible eliminar un proveedor que está siendo utilizado en alguna compra.

Herramientas administrativas

Desde el listado de compras, haciendo click sobre el enlace “Herramientas administrativas” se muestra el panel de herramientas de usuarios administradores tal como muestra la figura 6.16.



Figura 6.16: Compras: Herramientas administrativas

Estas herramientas permiten realizar operaciones de alto riesgo para la aplicación y permiten corregir situaciones anómalas del sistema.

Las herramientas administrativas del panel de compras, permiten:

- Gestionar tipos de compra. Esta opción permitirá al administrador, listar, crear, editar y eliminar los tipos de compra creados en la aplicación.
- Desamortizar compras. Esta opción revertirá el proceso de amortización mensual realizado automáticamente por el sistema a primero de mes. Devolviendo a cada compra el porcentaje y las cantidades amortizadas y eliminando los costes asumidos por cada paga de empleado del mes en curso.

- Amortizar compras. Esta opción emula el proceso automático de cambio de mes, amortizando el porcentaje correspondiente para cada compra en el mes en curso. Atención, esta opción tiene un alto riesgo, pues se pueden producir amortizaciones duplicadas, se deben haber desamortizado las compras para el mes en curso anteriormente.

6.2.3. Panel de Ventas

La sección dedicada a las ventas de la empresa se presenta en la Figura 6.17 y permite:

Listado de facturas emitidas

El listado de facturas emitidas presenta los datos más relevantes de la información de una factura emitida por la empresa, así como el cliente solicitante del producto o servicio, y el importe total de la factura.

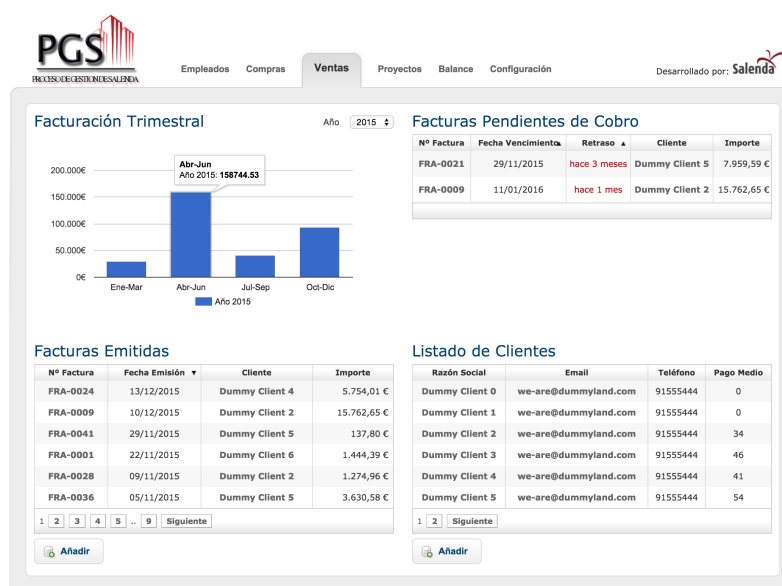


Figura 6.17: Panel de ventas

El listado de facturas puede ser ordenado por cualquiera de los campos que forman la tabla, pulsando en el título de cada columna. Cada pulsación invierte el orden de la columna (ascendente, descendente).

En la misma página se puede observar un gráfico comparativo que muestra la *facturación trimestral* de la empresa agrupada por años, presente en el gráfico 6.17.

Junto a este gráfico, encontramos un selector que permite cambiar el año del ejercicio económico, para poder revisar y comparar la facturación de los distintos ejercicios.

Listado de facturas pendientes de cobro

En el panel de ventas, junto al gráfico de *facturación trimestral* encontramos el listado de facturas *pendientes de cobro*. En este listado, solo aparecerán aquellas facturas cuya fecha de vencimiento sea anterior al día de hoy y su fecha de cobro sea nula.

El listado presenta los datos más relevantes para este tipo de factura como el n° de la factura, la fecha de vencimiento que tenía, el retraso expresado en forma natural (“hace 2 meses”, “hace 15 días”, ...), el cliente que se retrasó en el pago de la factura y el importe total pendiente.

Análogamente al listado de facturas emitidas, el listado de facturas pendientes puede ser ordenado por cualquiera de los campos que forman la tabla, pulsando en el título de cada columna. Cada pulsación invierte el orden de la columna (ascendente, descendente).

Generalmente, este listado aparecerá vacío, lo que supondrá que la facturación está al corriente de pagos.

Creación de factura y cliente. Adjuntar fichero

Haciendo click en el botón *Añadir* se despliega el formulario de creación de una factura tal como muestra la figura 6.18.

Una factura debe estar asociada a *un cliente* siempre (es posible crear y seleccionar un nuevo cliente desde el mismo formulario de creación de factura). Generalmente, también estará asociada a un proyecto, de esta manera se podrá llevar el cómputo de los ingresos totales producidos por un determinado proyecto de la empresa.

Para poder guardar una factura, es imprescindible especificar las fechas de emisión y vencimiento de la factura. Se puede especificar en el mismo momento la fecha de cobro si ya se conoce o ya ha acontecido. También se especificará el importe y la cuota de IVA de la factura, aunque para todos los cálculos internos de la aplicación se tendrá en cuenta

sólo el importe, ya que el IVA se debe gestionar siempre de manera independiente a la facturación de la empresa.

Figura 6.18: Creación de factura

Finalmente cada compra puede ir acompañada de un fichero adjunto. No existen restricciones de formato para estos ficheros, sólo se ha de configurar correctamente la ruta donde se alojarán los ficheros en el servidor.

Edición y eliminación de factura

Desde el listado de facturas, haciendo click sobre el *número* de una factura, se despliega el formulario de modificación de una factura reflejado en la figura 6.19.

Atención: la operación de eliminación de una factura no es reversible.

Listado de clientes

El listado de clientes presenta los datos más relevantes de la información de un cliente para el que la empresa realiza trabajos o presta servicios asiduamente, como puede observarse en la figura 6.17. El listado de clientes puede ser ordenado por cualquiera de los campos que forman la tabla, pulsando en el título de cada columna. Cada pulsación invierte el orden de la columna (ascendente, descendente).



Figura 6.19: Edición de factura

Edición y eliminación de cliente

Desde el listado de clientes, haciendo click sobre la *razón social* de un cliente, se despliega el formulario de modificación de un cliente reflejado en la figura 6.20. Atención: la operación de eliminación de un cliente no es reversible. No es posible eliminar un cliente que está siendo utilizado en alguna factura.



Figura 6.20: Edición de cliente

6.2.4. Panel de Proyectos

La sección dedicada a los proyectos de la empresa se presenta en la Figura 6.21 y permite:

Listado de proyectos

El listado de proyectos presenta los datos más relevantes de la información de un proyecto llevado a cabo por la empresa, así como cualquier prestación de servicios de la que se desee llevar traza de sus costes e ingresos, para finalmente deducir el beneficio global del proyecto.

Entre los campos del listado, los más relevantes son el *cliente* que solicita la realización del proyecto o es destino de la prestación de servicios; el *presupuesto* total del proyecto pactado a su inicio; el *coste* total acumulado del proyecto a día de hoy; el importe total *cobrado* a día de hoy; el *beneficio* neto resultante del proyecto a día de hoy; y finalmente un campo, que muestra el sumatorio total del *coste* de las horas invertidas en el proyecto por la empresa, para el *mes* en curso (este campo se ofrece como ayuda para facilitar la generación de la facturación mensual relativa a cada proyecto).



Figura 6.21: Panel de proyectos

El listado de proyectos puede ser ordenado por cualquiera de los campos que forman la tabla, pulsando en el título de cada columna. Cada pulsación invierte el orden de la columna (ascendente, descendente). Algunos de los campos son formulas y por lo tanto no podrán ser ordenados de ninguna manera.

Los campos del listado se calculan de manera dinámica cada 5 minutos, si desea un resultado actualizado inmediatamente se pueden actualizar los valores de manera interactiva pulsando el enlace “Actualizar valores” que aparece sobre el listado.

En la misma página se puede observar un gráfico comparativo que muestra el porcentaje de *proyectos agrupados por tiempo empleado* de los 6 proyectos con más porcentaje y la media del resto. Como muestra la figura 6.22.

Junto a este gráfico, encontramos un selector que permite cambiar el criterio de agrupación de la gráfica para observar el porcentaje de *proyectos agrupados por el cliente* que solicita el proyecto o la prestación de servicios.

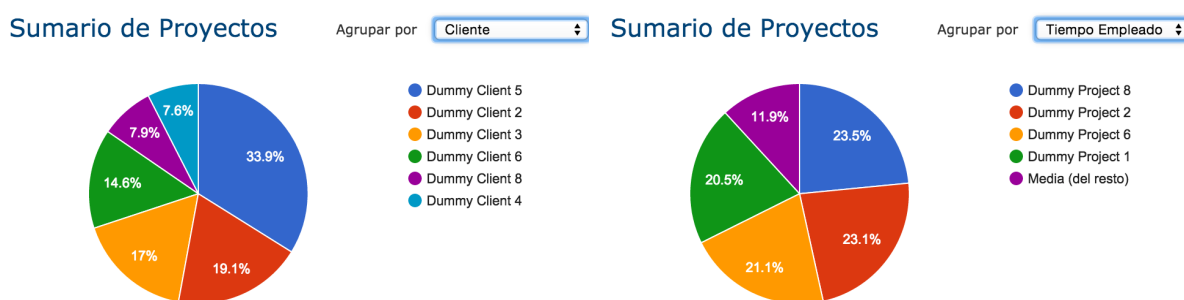


Figura 6.22: Grafica de proyectos. Sumario de proyectos

Junto a éste, otro gráfico muestra el porcentaje de dinero *presupuestado vs cobrado* de todos los proyectos o prestaciones de servicios de la empresa. Como muestra la figura 6.23.

Junto a este gráfico, encontramos un selector que permite cambiar el criterio de agrupación de la gráfica para observar el porcentaje de dinero que supone un *beneficio vs coste* para la empresa.

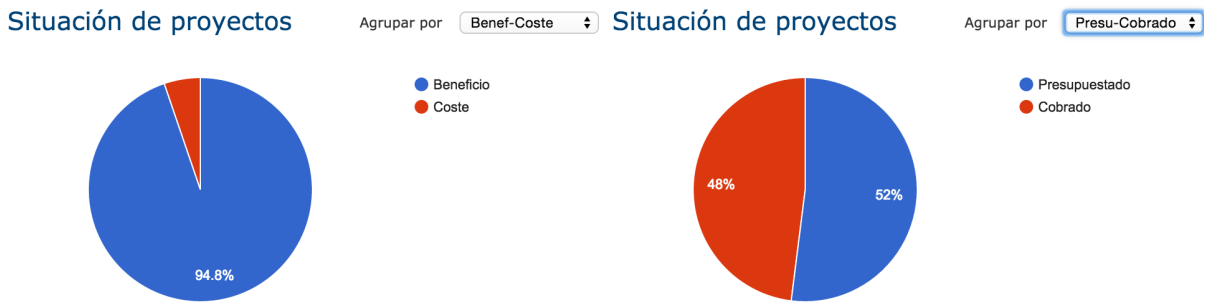


Figura 6.23: Grafica de proyectos. Situación de proyectos

Creación de proyecto

Haciendo click en el botón “Añadir” se despliega el formulario de creación de un proyecto tal como muestra la figura 6.24.

Desde él, se puede especificar el nombre del proyecto, el cliente que lo solicita (Opcional) y el presupuesto pactado para el proyecto. Se puede crear un nuevo cliente desde este mismo formulario y seleccionarlo para este proyecto.

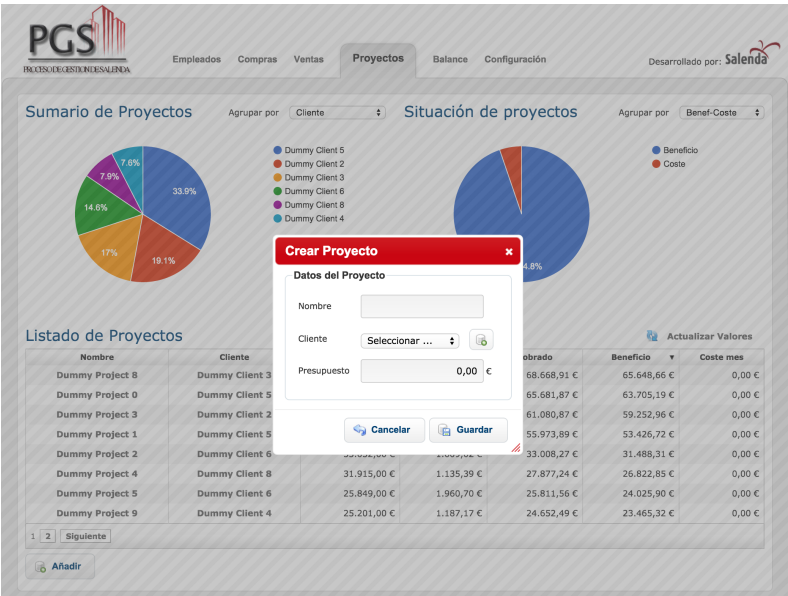


Figura 6.24: Creación de proyecto

Edición y eliminación de proyecto

Haciendo click en el *nombre* de un proyecto en el listado de proyectos, se despliega el formulario de modificación de un proyecto reflejado en la figura 6.25, desde el cual se pueden cambiar sus datos, o eliminarlo definitivamente. También se pueden observar su coste y beneficio o modificar el total presupuestado y cobrado. Se recomienda no modificar el total cobrado salvo necesidad de ajuste contable, ya que este dato se calcula de manera automática con cada factura emitida asociada a este proyecto.

Atención: la operación de eliminación de un proyecto no es reversible. No es posible eliminar un proyecto que está siendo utilizado en alguna factura o tiene worklogs.

Detalles de Proyecto

Datos del Proyecto

Nombre: Dummy Project 8 Coste: 3035,37 €

Cliente: Dummy Client 3 Cobrado: 68668,91 €

Presupuesto: 68704,00 € Beneficio: 65648,66 €

Últimos Worklogs

Issue Key	Fecha	Técnico	Tiempo Empleado	beneficio	Coste mes
DP40-75	30/08/2015	Dumb 4 Dummy Dummy	1d 5h 24m	65.648,66 €	0,00 €
DP40-79	27/08/2015	Dumb 8 Dummy Dummy	1d 1h 36m	63.705,19 €	0,00 €
DP40-10	26/08/2015	Dumb 4 Dummy Dummy	2d 4h 47m	59.252,96 €	0,00 €
				53.426,72 €	0,00 €
				31.488,31 €	0,00 €
				26.822,85 €	0,00 €
				24.025,90 €	0,00 €
				23.465,32 €	0,00 €

Figura 6.25: Edición de proyecto

Listado de worklogs

Desde la pantalla de modificación de un proyecto presente en la figura 6.25 también se accede al listado de “Últimos Worklogs” del proyecto. Estos worklogs, son reportes de trabajo de los empleados en un determinado proyecto, en una fecha y con una duración.

La cantidad de worklogs de un proyecto afecta de manera directa a su coste, pues éste es el resultante de multiplicar el número de horas empleadas por un empleado en

el proyecto por el coste del empleado, teniendo en cuenta la fecha en la que aconteció el worklog y por lo tanto el precio mensual del empleado que le afecta.

El listado de worklogs puede ser ordenado por cualquiera de los campos que forman la tabla, pulsando en el título de cada columna. Cada pulsación invierte el orden de la columna (ascendente, descendente).

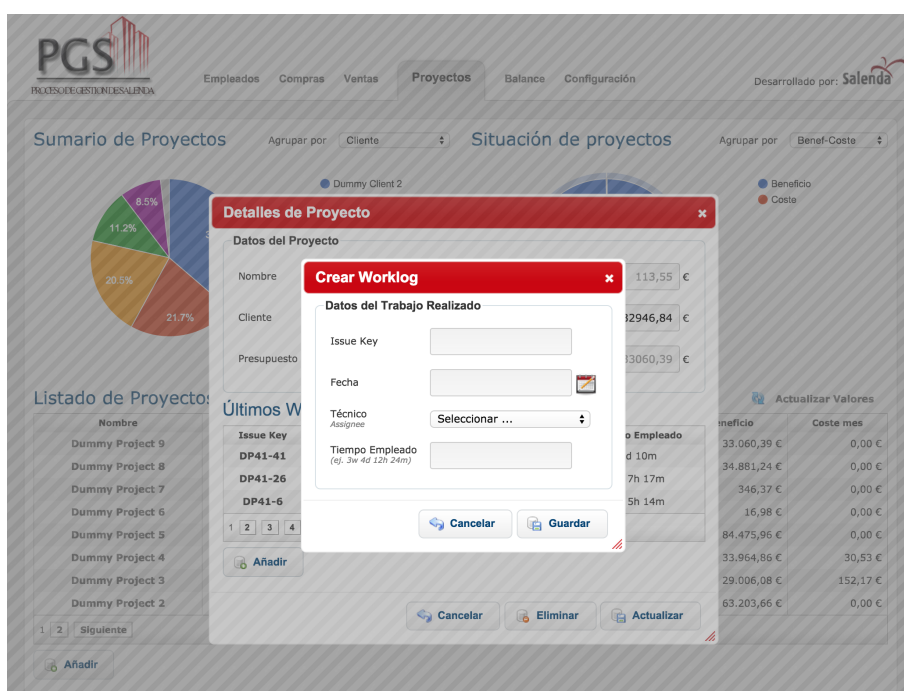


Figura 6.26: Creación de worklog

Creación de worklogs

Haciendo click en el botón *Añadir* se despliega el formulario de creación de un worklog tal como muestra la figura 6.26.

Desde él, se puede especificar la información del worklog, su clave única, su fecha, el técnico que realizó el trabajo y el tiempo empleado. El tiempo empleado puede ser introducido mediante pseudo-lenguaje, utilizando las siglas de los periodos temporales, a saber: s (segundos), m (minutos), h (horas), d (días) y w (semanas).

Con cada nuevo worklog, la aplicación recalcula los costes del proyecto en el que se introduce el reporte de trabajo.

Edición y eliminación worklogs

Desde el listado de worklogs, haciendo click sobre la *clave* de un worklog, se despliega el formulario de modificación de un worklog reflejado en la figura 6.27:

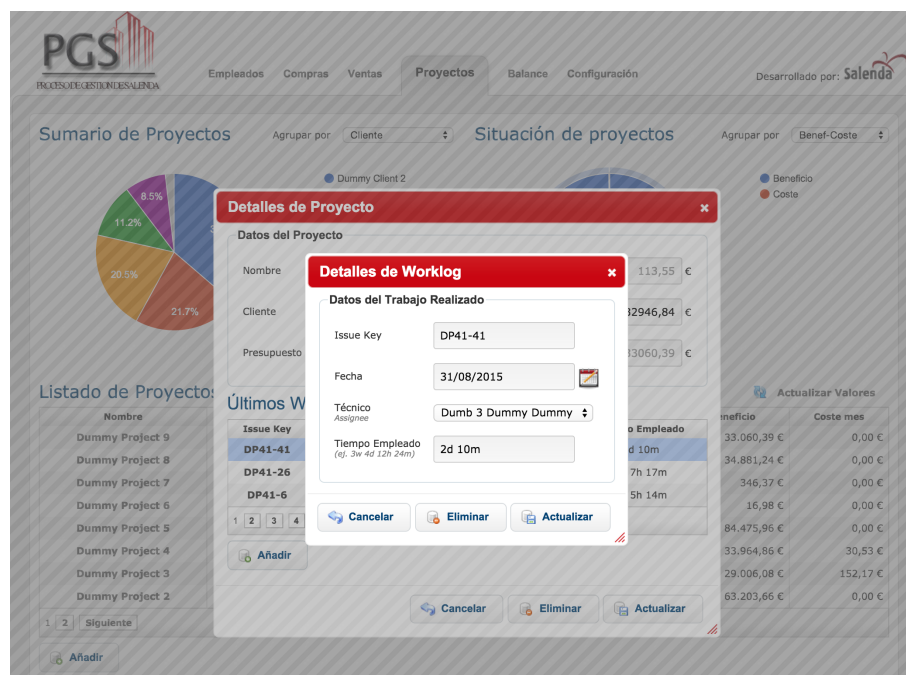


Figura 6.27: Edición de worklog

Atención, la operación de eliminación de un worklog no es reversible. Si se elimina o modifica un worklog, se verán afectados los costes del proyecto al que pertenece.

6.2.5. Panel de Balance

La sección dedicada al balance de caja de la empresa se presenta en la Figura 6.28 y permite:

Previsión de caja

El panel de balance presenta en su parte superior, un gráfico temporal de la variación del saldo de caja de la empresa, tal como muestra la figura 6.28. En éste, se presentan de manera cronológica las *transacciones de caja* (los “pagos” y “cobros”) a los que ha hecho, o debe hacer frente la empresa.

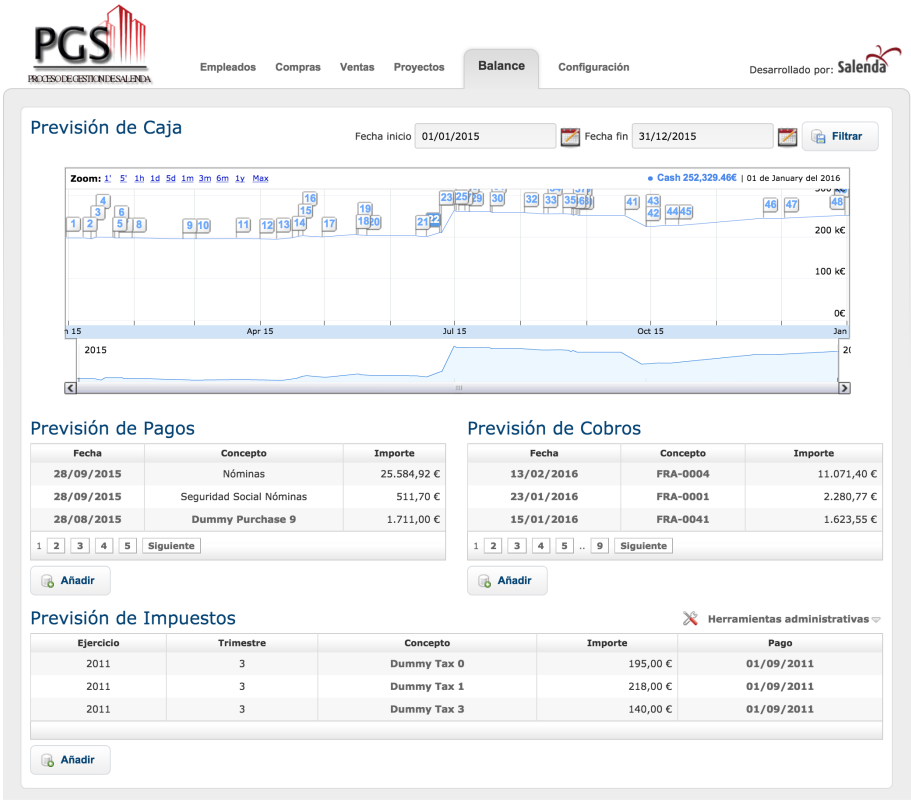


Figura 6.28: Panel de Balance

Este gráfico, muestra las transacciones de caja filtradas por fecha, incluyendo por defecto aquellas cuya fecha esta dentro del año en curso. Opcionalmente, se puede filtrar el gráfico por cualquier combinación de fechas utilizando los selectores *fecha inicio* y *fecha fin*. Análogamente, el gráfico permite la posibilidad de hacer *zoom*, en un rango de fechas específico, sin necesidad de filtrar las transacciones. Este *zoom* permite desplazarse por el gráfico, observando en detalle cualquier sucesión de operaciones en el tiempo, dentro del conjunto filtrado por los selectores de fecha superiores. Haciendo click en cualquiera de las etiquetas que representan cada transacción, se despliega la ventana del detalle del origen de la transacción, ya sea una compra, una factura, un cobro o un pago, como muestra la figura 6.29.

Previsión de pagos

El listado de previsión de pagos presenta los datos más relevantes de la información de un pago al que tiene que hacer frente la empresa y puede observarse en el gráfico 6.28.

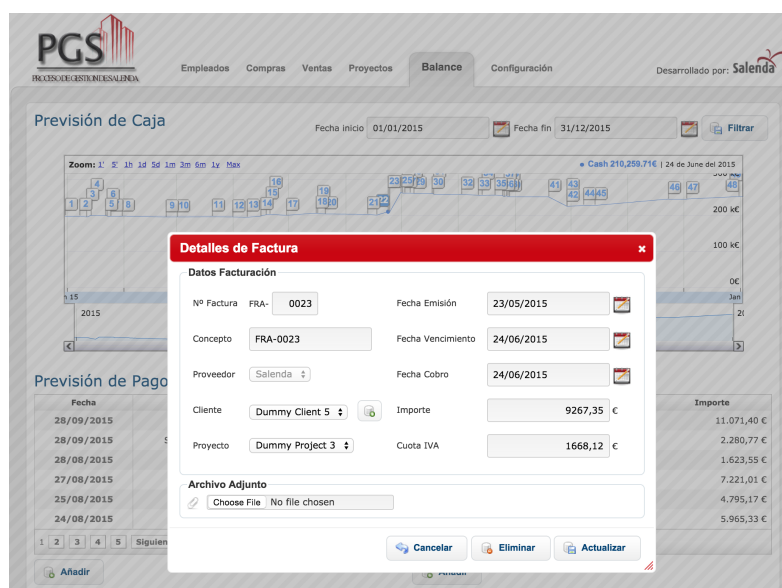


Figura 6.29: Panel de Balance: detalle

El listado de previsión de pagos puede ser ordenado por cualquiera de los campos que forman la tabla, pulsando en el título de cada columna. Cada pulsación invierte el orden de la columna (ascendente, descendente).

Creación de pago

Haciendo click en el botón *Añadir* se despliega el formulario de creación de un pago tal como muestra la figura 6.30.

Generalmente, los transacciones de pago, son generadas de manera automática por la aplicación, cada vez que la empresa incurre en un gasto (emisión de nóminas mensual, cuotas de la seguridad social, compras, etc.). Cada una de estas operaciones tendrá de manera automática una transacción de pago asociada, pero, es posible introducir en el sistema pagos independientes a otras operaciones si es necesario. Estas transacciones serán reflejadas en el gráfico de previsión de caja como un gasto más.

Modificación y eliminación de pago

Desde el listado de previsión de pagos, haciendo click sobre la *fecha* de un pago, se despliega el formulario de modificación de un pago reflejado en la figura 6.31

Atención: la operación de eliminación de un pago no es reversible. No es posible eliminar un pago asociado a una operación de compra.

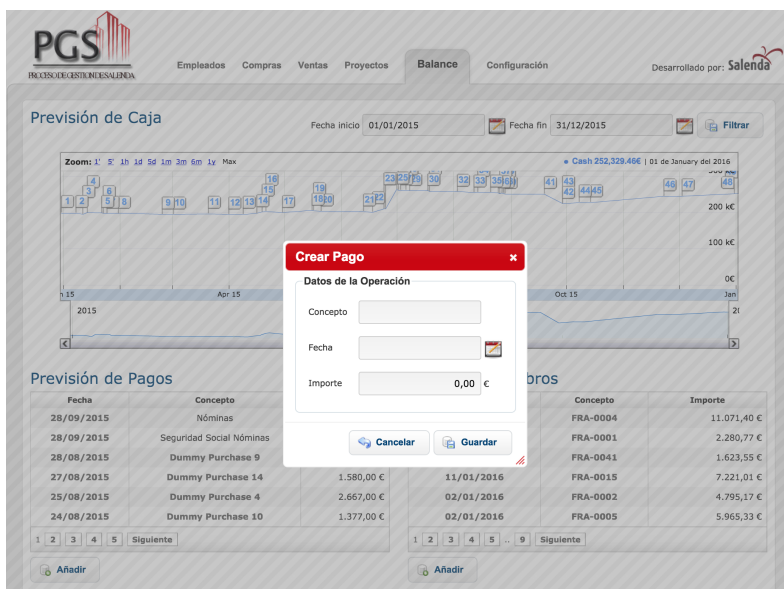


Figura 6.30: Creación de pago

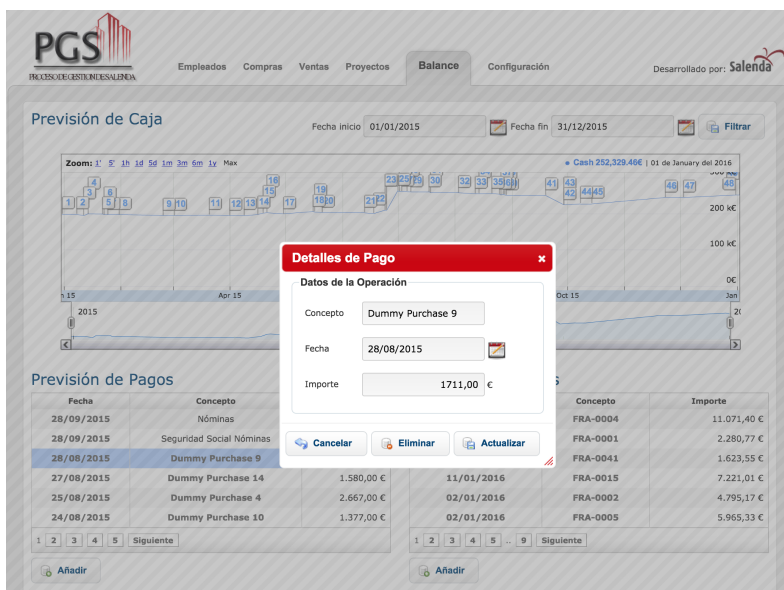


Figura 6.31: Modificación de pago

Previsión de cobros

El listado de previsión de cobros presenta los datos más relevantes de la información de un cobro que la empresa tiene previsto hacer y puede observarse en el gráfico 6.28.

El listado de previsión de cobros puede ser ordenado por cualquiera de los campos que forman la tabla, pulsando en el título de cada columna. Cada pulsación invierte el orden de la columna (ascendente, descendente).

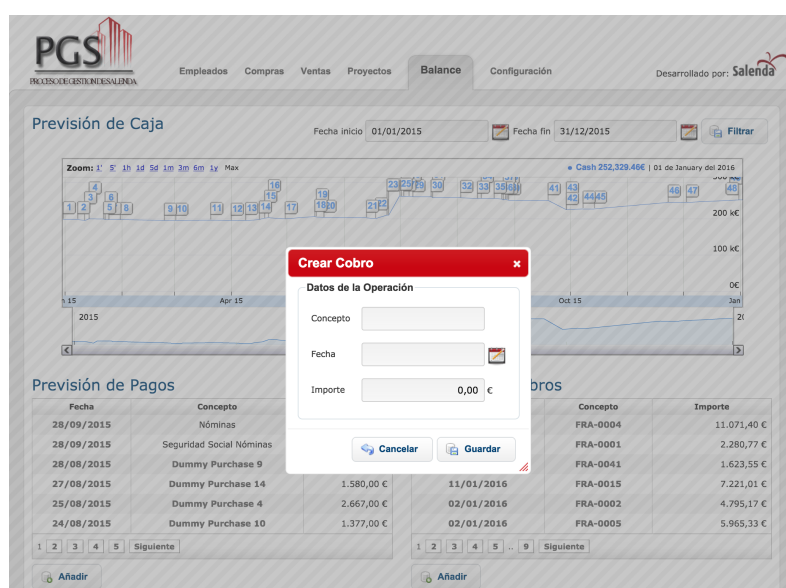


Figura 6.32: Creación de cobro

Creación de cobro

Haciendo click en el botón *Añadir* se despliega el formulario de creación de un cobro tal como muestra la figura 6.32.

Generalmente, las transacciones de cobro, son generadas de manera automática por la aplicación, cada vez que la empresa emite una factura por la prestación de un servicio, o la realización de trabajos en un determinado proyecto. Cada una de estas facturas tendrá de manera automática una transacción de cobro asociada, pero, es posible introducir en el sistema cobros independientes a facturas si es necesario. Estas transacciones serán reflejadas en el gráfico de previsión de caja como un ingreso más.

Modificación y eliminación de cobro

Desde el listado de previsión de cobros, haciendo click sobre la *fecha* de un cobro, se despliega el formulario de modificación de un cobro reflejado en la figura 6.33.

Atención: la operación de eliminación de un cobro no es reversible. No es posible eliminar un cobro asociado a una factura de venta.

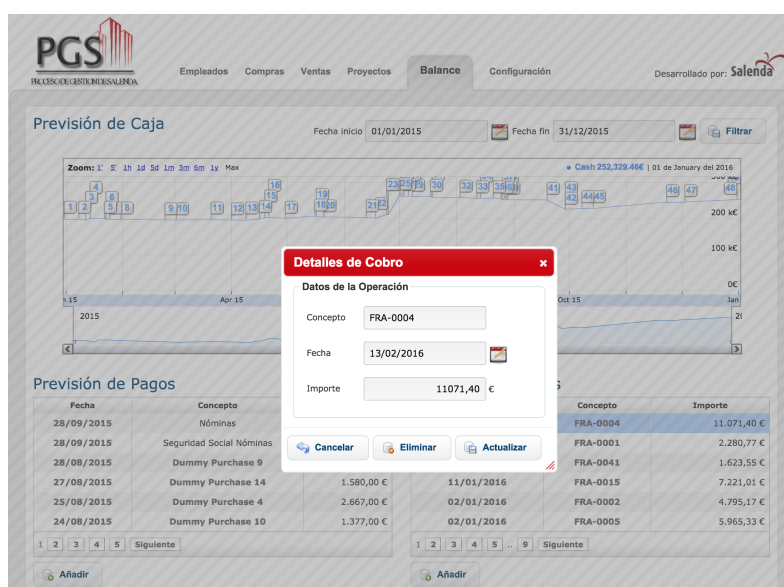


Figura 6.33: Modificación de cobro



Figura 6.34: Creación de impuesto

Previsión de impuestos

El listado de previsión de impuestos presenta los datos más relevantes de la información de un impuesto a la que la empresa tiene que hacer frente y puede observarse en el gráfico 6.28.

El listado de previsión de impuestos puede ser ordenado por cualquiera de los campos que forman la tabla, pulsando en el título de cada columna. Cada pulsación invierte el orden de la columna (ascendente, descendente).

Creación de impuesto

Haciendo click en el botón *Añadir* se despliega el formulario de creación de un impuesto tal como muestra la figura 6.34.

Cuando se crea un impuesto, se debe especificar, el ejercicio fiscal y el trimestre al que está afecto el impuesto, su importe y el concepto. Con cada nuevo impuesto, la aplicación generará una transacción de pago coincidiendo con la fecha del impuesto.

Modificación y eliminación de impuesto

Desde el listado de previsión de impuestos, haciendo click sobre el *concepto* de un impuesto, se despliega el formulario de modificación de un impuesto reflejado en la figura 6.35



Figura 6.35: Modificación de impuesto

Atención: la operación de eliminación de un impuesto no es reversible. No es posible

eliminar un pago asociado a un impuesto.

Herramientas administrativas

Desde el listado de previsión de impuestos, haciendo click sobre el enlace “Herramientas administrativas” se muestra el panel de herramientas de usuarios administradores tal como muestra la figura 6.36.

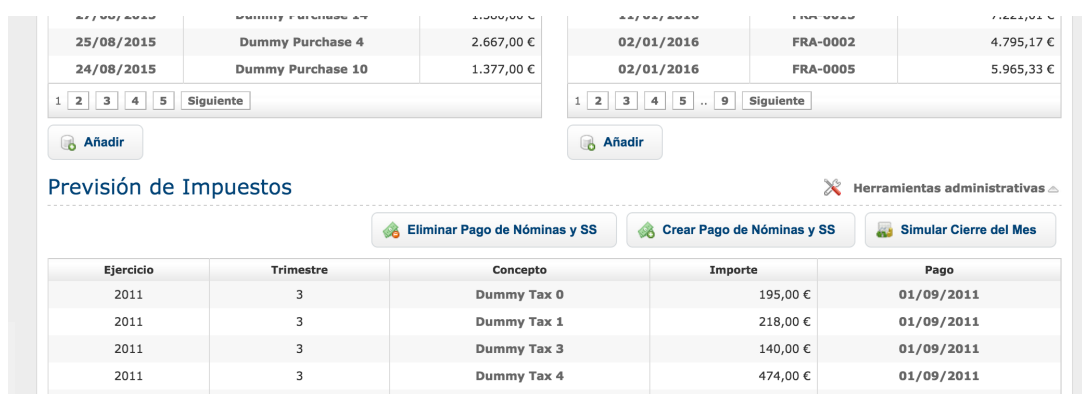


Figura 6.36: Balance: Herramientas administrativas

Estas herramientas permiten realizar operaciones de alto riesgo para la aplicación y permiten corregir situaciones anómalas del sistema.

Las herramientas administrativas del panel de balance, permiten:

- Eliminar pago de nóminas y SS. Esta opción permitirá al administrador, eliminar las transacciones de pago de nóminas y cuotas de la SS, calculadas por la aplicación de manera automática en el proceso de cambio de mes.
- Crear pago de nóminas y SS. Esta opción emula parte del proceso automático de cambio de mes, generando las transacciones de pago de nóminas y cuotas de la SS, para los empleados en estado activo con pagas para el mes en curso.
- Simular cierre del mes. Esta opción emula el proceso automático de cambio de mes, realizando todas las operaciones que se llevan a cabo a primero de mes: copiando las pagas del mes anterior, amortizando todas las compras, generando las transacciones de pago de nóminas y cuotas de la SS y copiando las compras recurrentes.

6.2.6. Panel de Configuración

La sección dedicada a la configuración de la aplicación se presenta en la Figura 6.37 y permite:

Figura 6.37: Panel de configuración

Configuración de empresa usuario

En este apartado se puede seleccionar la empresa que actuará como usuario de la aplicación. Actualmente, solo una empresa puede utilizar al mismo tiempo la aplicación, aunque es posible modificar los datos de ésta, o crear una nueva e iniciar su uso desde el principio.

Configuración del servidor

En este apartado se puede especificar la ruta de almacenamiento del gestor documental. Esta ruta debe tener los permisos adecuados para que la aplicación pueda escribir en ella (usuario “tomcat”, grupo “tomcat”, y permisos de escritura son necesarios).

Junto a la ruta, se puede utilizar el botón “comprobar” que tratará de crear un fichero de pruebas y guardarlo para comprobar que los permisos de escritura están correctamente

configurados. Este fichero será borrado después de la ejecución de esta acción.

Configuración del tiempo

En este apartado se puede especificar todos los periodos de tiempo que personalizan el modelo de ejecución de la aplicación. Así se puede decidir las duraciones de conceptos de la lógica de negocio tales como la duración de la *semana laboral*, de la *jornada laboral*, las *horas laborables de cada mes*, etc...

También se puede especificar la cadencia de actualización de los valores calculados de los proyectos.

Finalmente, dado que en cada región o país puede ocurrir, que los periodos fiscales sean diferentes, se permite personalizar las fechas de fiscalización de los trimestres del ejercicio anual. Estas son las fechas que tomaran las operaciones de pago asociadas a pagos de impuestos en cada trimestre.

Capítulo 7

Conclusiones y Líneas Futuras

En este capítulo, se enumeran las conclusiones, los objetivos alcanzados y las experiencias, tanto personales como profesionales obtenidas de la realización de este proyecto y las líneas futuras de desarrollo que quedan pendientes de implementación.

7.1. Conclusiones

Una vez finalizado el desarrollo del proyecto, podemos analizar si se han cumplido o no los objetivos planteados al inicio, y cómo el camino hasta alcanzarlos ha influido personal y profesionalmente al autor.

7.1.1. Herramienta de gestión de proyectos de software

El primer objetivo que se perseguía, era desarrollar una herramienta, que ayudara a la gestión de proyectos de software, aportando información que representara una ventaja competitiva en un mercado muy explotado. Este objetivo ha sido cumplido, pues se desarrolló una aplicación que cubría todas las especificaciones funcionales esperadas por cliente, se desplegó en producción y actualmente sigue en uso.

Esta aplicación, no solo ha conseguido unificar funcionalidades, que se encontraban distribuidas en otras tres herramientas de software, sino que el hecho de haber unificado esas funcionalidades en una misma herramienta de software y en una misma base de datos, ha permitido explotar toda esa información para generar nuevas métricas que aportan información de relevancia y valor para una empresa de desarrollo de software. Entre otras funcionalidades, PGS ha conseguido alcanzar los siguientes objetivos funcionales relevantes:

- Gestión de salarios y costes de empleados.
- Cálculo del coste/hora y horas/mes, en tiempo real, de cada empleado.
- Identificación de empleados más productivos.
- Identificación de empleados más trabajadores.
- Gestión de compras, facturas, amortización de bienes muebles simplificada.
- Distribución de costes sobre los empleados y proyectos de la empresa, para poder repercutir esos gastos en la facturación posteriormente de una manera distribuida y razonable.
- Gestión de costes y beneficios de proyectos en tiempo real, mejor llegada al mercado, más competitivos.

- Presupuestos y facturación realistas, adaptados a los costes de la propia empresa: sostenibilidad y crecimiento de la empresa.
- Conocimiento de la situación financiera de la empresa de servicios (no siempre es tangible).
- Previsión de ingresos y costes: anticipación y reacción pronta.

Mediante el uso de esta aplicación, Salenda [1], ha conseguido mejorar su facturación, ofreciendo costes acordes a sus gastos y con un margen de beneficios suficiente y sostenible, que le permite competir de manera ventajosa en su mercado.

Personalmente, el proceso de realización de este desarrollo, y sobretodo, la labor de análisis y diseño previas a la implementación, me han permitido conocer, de una manera muy temprana, todos los procesos de gestión de una mediana empresa de desarrollo: desde cómo afectan los costes de los recursos humanos o las compras y gastos que tiene una empresa, al origen de los importes de la facturación, cómo se amortizan los bienes materiales e inmateriales de una empresa, las obligaciones impositivas que tiene o la compleja labor de financiación que deben afrontar las pymes españolas para mantener su flujo de caja en positivo, o incluso con suerte, poder llegar a generar beneficios.

7.1.2. Herramientas ágiles de desarrollo

El segundo objetivo que se perseguía, era optimizar al máximo los recursos destinados al proyecto, tanto económicos como en tiempo, al tratarse de un desarrollo de financiación propia y para explotación interna de la empresa. Para esto, se analizó la situación de las herramientas y metodologías de desarrollo de software existentes en el momento de realización del proyecto, llegando a la conclusión, junto con la empresa cliente, de que debían ser Grails y Scrum, el framework y la metodología de desarrollo a utilizar respectivamente.

Estos dos frameworks (de implementación y de planificación) han permitido, llevar a cabo un proyecto con una funcionalidad extensa, en solo 4 meses de desarrollo, incluyendo análisis e implementación, por 1 sola persona a tiempo completo; cuando, según comentó la empresa cliente, se había planificado en ocasiones anteriores para que fuera realizado por 2 personas a tiempo completo (utilizando otros frameworks java anteriores como Struts y metodologías más pesadas), en un tiempo no inferior a 6 meses.

Personalmente, desarrollar con un framework ágil como Grails ha supuesto para mí, la motivación última y decisiva para dedicarme a esta profesión, con una ilusión antes no encontrada. Todas las ayudas y soluciones que Grails aporta al desarrollador, le permiten alejarse de las limitaciones asociadas al entorno de desarrollo, para dedicarse a lo que realmente es su vocación, la *Ingeniería de Software*, la resolución de problemas complejos de una manera eficaz y óptima, sin necesidad de tener que bregar y sufrir las vicisitudes y limitaciones de sus herramientas de trabajo.

Tanto es así, que los últimos 5 años, he estado trabajando (en Salenda y otras empresas), siempre teniendo este framework como mi herramienta de trabajo diaria, y me ha motivado para participar activamente en la comunidad groovy & grails, realizando charlas como ponente en eventos y participando en los foros de la comunidad.

Análogamente, el descubrimiento de las metodologías ágiles de desarrollo, han supuesto para mí, un cambio en el entendimiento de los procesos de desarrollo del software, alejándome de la idea de la documentación extensa y la definición pronta de requisitos. La capacidad de adaptación al cambio, la flexibilidad y rapidez de respuesta, se han convertido en los pilares de mi proceso de desarrollo, y fueron estas características las que permitieron completar un desarrollo como PGS en un tiempo inferior al esperado.

Particularmente, Scrum, me ha convencido como una de las mejores implementaciones de metodologías ágiles. Junto con Kanban, son las principales metodologías que utilizo en mi vida profesional. Fue el flujo de trabajo de Scrum, el que permitió mediante iteraciones de 2 semanas, llevar a cabo el desarrollo, incluyendo, análisis, implementación y tests de cada módulo de funcionalidad en una única iteración (por cada módulo). La flexibilidad y la capacidad de adaptación a los cambios fue determinante, para poder llevar cabo el proyecto, ya que aunque se tenía una idea clara de los objetivos, no se tenían todos los requisitos definidos al inicio del proyecto. Sin embargo, la falta de necesidad de documentación previa a la implementación que propone Scrum, supuso un trabajo extra a la finalización del mismo, por lo tanto, dependiendo del marco en el que se vaya a implementar el proyecto, podría ser beneficioso incluir una etapa de documentación en el sprint de scrum (aunque esto no sea lo recomendado por la metodología).

Existía un tercer objetivo oculto entre los 2 anteriores, que la empresa Salenda perseguía, y se alcanzó prácticamente de manera automática: el aprendizaje y familiarización

del ingeniero proyectista con los frameworks Grails y Scrum, pilares del proceso de desarrollo de la empresa Salenda, en un tiempo razonablemente corto, pues a la finalización del desarrollo, fui incorporado a la plantilla del departamento de desarrollo de la empresa, como Ingeniero de Software, realizando labores de desarrollo e impartiendo cursos de formación en metodologías ágiles y de programación con el framework Grails para otras empresas.

Por lo tanto, tanto los conocimientos adquiridos a lo largo de la carrera de Ingeniería Técnica Informática de Gestión, como la realización última de este proyecto, me han permitido incorporarme de una manera rápida al mercado laboral, incluso teniendo en cuenta que la situación económica y laboral del país en el año en que se realizó este proyecto, no era la más favorable.

7.2. Líneas Futuras

Aunque el desarrollo, se encuentra en producción y bajo uso cotidiano de la empresa, se dejaron abiertas ciertas líneas de desarrollo, que surgieron durante la implementación, como posible ampliación o mejora de la funcionalidad. Algunas de ellas, fueron desarrolladas posteriormente dada su demandada necesidad.

- Seguridad: Creación de usuarios, roles y perfiles para el acceso y manejo de la herramienta.

Resultó evidente, durante el desarrollo de la herramienta, que sería necesario algún tipo de control de acceso y personalización, de las capacidades de interacción del usuario mediante algún tipo de herramienta de gestión de roles y permisos de acceso. Sin embargo, dado que la aplicación iba a ser desplegada dentro de la propia vpn de la empresa, no fue un requisito imprescindible de la etapa inicial del desarrollo.

- Automatización de ingesta de datos: worklogs

Poco después de los primeros días de uso de la herramienta, comenzó a ser evidente, que la tarea manual de reporte de los informes de trabajo por parte de los empleados, suponía un esfuerzo doble, ya que en ningún momento, se dejó de utilizar Jira como la herramienta de seguimiento de incidencias principal. Por esto, surgió la idea, de

desarrollar un plugin de grails, que permitiera a PGS realizar consultas al sistema Jira, utilizando su API REST, para ingestar todas las incidencias y reportes de trabajo, e incluso los proyectos dados de alta, de manera automática, simplificando mucho las labores de los usuarios de PGS.

- Gestor documental más potente

Aunque la herramienta permite adjuntar ficheros a las facturas del sistema, resultó una funcionalidad limitada, ya que no resolvía tareas cotidianas, como la búsqueda de una determinada factura, por el texto que esta contenía, o la capacidad de descargar de una sola vez, todas las facturas de un determinado mes. Por esto, se propuso como mejora, la integración de un servidor documental como Alfresco, que da solución a todas las necesidades anteriores y otras muchas.

- Generación de facturas en pdf con plantillas

Aunque la herramienta permitía, generar facturas en el sistema, aportando toda la información necesaria para poder representar un flujo de caja fiel a la situación financiera real de la empresa, se encontró la necesidad no cubierta de generar las facturas en papel, con un formato y diseños acordes a la imagen de la empresa. Por esto, se propuso como mejora la integración de un sistema de generación de informes automático como JasperReports o iReports, ambas herramientas java para la generación de informes en PDF, utilizando plantillas como base del documento.

- Emisión de transferencias

Otra mejora propuesta, fue la integración con las APIS bancarias, para poder emitir desde la herramienta, pagos a los empleados y a los proveedores de la empresa, o cargos a los clientes.

- Ingesta de datos bancarios para balance

Análogamente al caso anterior, se propuso integrar las APIS bancarias en el sistema, para ingestar todos los movimientos de las cuentas de la empresa y poder reflejarlos en el módulo de balance, que en conjunto con las operaciones previstas, darían un reflejo no solo de la situación actual del balance de caja, sino una previsión muy ajustada del futuro de la misma.

- Gestión de comunicaciones:

Resultó de gran ayuda, tener un pequeño CRM de Clientes y Proveedores con los que la empresa trabajaba, sin embargo, surgió la idea de poder tener el histórico de comunicaciones con estos clientes y proveedores dentro de la herramienta, para que pudiera ser consultado por el resto de usuarios y dar un soporte más personalizado a los mismos.

Bibliografía

- [1] Salenda. <http://www.salenda.es>.
- [2] Pyme: Pequeña y mediana empresa.
http://es.wikipedia.org/wiki/Pequena_y_mediana_empresa.
- [3] Gestión de proyectos. http://es.wikipedia.org/wiki/Gestion_de_proyectos.
- [4] Framework. <http://es.wikipedia.org/wiki/Framework>.
- [5] Grails. <https://grails.org>.
- [6] Jvm: Java virtual machine.
http://es.wikipedia.org/wiki/Maquina_virtual_Java.
- [7] Groovy. <http://groovy.codehaus.org>.
- [8] Convención sobre configuración.
http://es.wikipedia.org/wiki/Convencion_sobre_Configuracion.
- [9] Metodología de desarrollo de software.
http://es.wikipedia.org/wiki/Metodologia_de_desarrollo_de_software.
- [10] Scrum. <http://es.wikipedia.org/wiki/Scrum>.
- [11] Ide: Entorno de desarrollo integrado. http://es.wikipedia.org/wiki/Entorno_de_desarrollo_integrado.
- [12] Gts/ggts: Groovy/grails tool suite. <https://grails.org/products/ggts>.

- [13] Svn: Subversion. sistema de control de versiones. [http://es.wikipedia.org/wiki/Subversion_\(software\)](http://es.wikipedia.org/wiki/Subversion_(software)).
- [14] Apache tomcat. servidor de aplicaciones. <http://tomcat.apache.org/>.
- [15] Mysql. servidor de bases de datos. <http://www.mysql.com/>.
- [16] Atlassian jira. software de gestión de proyectos. <https://es.atlassian.com/software/jira>.
- [17] Alvaro sanchez-mariscal. <https://www.linkedin.com/in/mariscal>.
- [18] Universidad carlos iii de madrid. <http://www.uc3m.es/Home>.
- [19] Gul. grupo de usuarios de linux. universidad carlos iii de madrid. <http://www.gul.es/>.
- [20] Centro de atencion a usuarios de la universidad carlos iii de madrid. <http://portal.uc3m.es/portal/page/portal/informatica/CAU>.
- [21] Imagina works. <http://www.imaginaworks.com/>.
- [22] Escuela de groovy. <http://www.escueladegroovy.com>.
- [23] Roberto martín flores. <https://www.linkedin.com/in/robertomartinflores>.
- [24] J2ee, java ee. http://es.wikipedia.org/wiki/Java_EE.
- [25] Release life cycle.
http://en.wikipedia.org/wiki/Software_release_life_cycle.
- [26] Sugarcrm. <http://www.sugarcrm.com/es>.
- [27] Crm. customer relationship management.
http://es.wikipedia.org/wiki/Customer_relationship_management.
- [28] Lamp. linux, apache, mysql & php. <http://es.wikipedia.org/wiki/LAMP>.
- [29] Return of investment. http://en.wikipedia.org/wiki/Return_on_investment.
- [30] Gmail. <http://es.wikipedia.org/wiki/Gmail>.

- [31] Twitter. <http://es.wikipedia.org/wiki/Twitter>.
- [32] Gotomeeting. <http://en.wikipedia.org/wiki/GoToMeeting>.
- [33] Google docs(aka google drive). http://es.wikipedia.org/wiki/Google_Drive.
- [34] Sage. <http://www.sage.es/cabecera/sobre-sage/la-empresa>.
- [35] Manifesto for software craftsmanship. <http://manifesto.softwarecraftsmanship.org/>.
- [36] Clean code by uncle bob. <http://cleancoders.com/>.
- [37] Test driven development. http://es.wikipedia.org/wiki/Desarrollo_guiado_por_pruebas.
- [38] Java. https://www.java.com/es/download/whatis_java.jsp.
- [39] Sun microsystems. http://es.wikipedia.org/wiki/Sun_Microsystems.
- [40] Oracle. <http://www.oracle.com/es/index.html>.
- [41] Enterprise java beans. http://es.wikipedia.org/wiki/Enterprise_JavaBeans.
- [42] Java message service. http://es.wikipedia.org/wiki/Java_Message_Service.
- [43] Java servlet. http://es.wikipedia.org/wiki/Java_Servlet.
- [44] Javaserer faces. http://es.wikipedia.org/wiki/JavaServer_Faces.
- [45] Open source. código abierto. http://en.wikipedia.org/wiki/Open_source.
- [46] Application programming interface.
http://en.wikipedia.org/wiki/Application_programming_interface.
- [47] Java database connectivity. http://es.wikipedia.org/wiki/Java_Database_Connectivity.
- [48] Lenguajes dinámicos. http://en.wikipedia.org/wiki/Dynamic_programming_language.

- [49] Lenguajes estáticos.
http://es.wikipedia.org/wiki/Lenguaje_de_programacion.
- [50] Closures. [http://en.wikipedia.org/wiki/Closure_\(computer_programming\)](http://en.wikipedia.org/wiki/Closure_(computer_programming)).
- [51] Lambdas, funciones anónimas.
http://en.wikipedia.org/wiki/Anonymous_function.
- [52] G2one. <https://www.crunchbase.com/organization/g2one>.
- [53] Springsource. <http://en.wikipedia.org/wiki/SpringSource>.
- [54] Springframework. http://en.wikipedia.org/wiki/Spring_Framework.
- [55] Hibernate. <http://es.wikipedia.org/wiki/Hibernate>.
- [56] Agile manifesto. http://en.wikipedia.org/wiki/Agile_software_development#The_Agile_Manifesto.
- [57] Scrum roles. scrum pocket edition slides. <http://www.slideshare.net/vatemu/cas2-k14-scrumpocketedition>.
- [58] Poc. http://es.wikipedia.org/wiki/Prueba_de_concepto.
- [59] Netbeans. <http://es.wikipedia.org/wiki/NetBeans>.
- [60] Eclipse. <https://eclipse.org/ide/>.
- [61] Jdk. http://es.wikipedia.org/wiki/Java_Development_Kit.
- [62] Jetty. <http://es.wikipedia.org/wiki/Jetty>.
- [63] Hsqldb. <http://es.wikipedia.org/wiki/HSQLDB>.
- [64] Grailsdoc. <http://grails.github.io/grails-doc/1.3.7/guide/single.html>.
- [65] Scaffolding. http://en.wikipedia.org/wiki/Scaffold_%28programming%29.
- [66] Mvc. modelo-vista-controlador. <http://es.wikipedia.org/wiki/Modelo%E2%80%9393vista%E2%80%9393controlador>.

-
- [67] Gorm. <http://grails.github.io/grails-doc/3.0.x/guide/GORM.html>.
- [68] Data access object. http://es.wikipedia.org/wiki/Data_Access_Object.